



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
INFORMÁTICA E ESTATÍSTICA**

RAFAEL CRISPIM IGNÁCIO

GUIA FACETADO DE TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

Florianópolis

2018

RAFAEL CRISPIM IGNÁCIO

GUIA FACETADO DE TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

Trabalho de conclusão de curso submetido ao curso de Sistemas de Informação para a obtenção do Grau de Bacharel em Sistemas de Informação .

Orientadora: Profa. Dra. Fabiane Barreto Vavassori Benitti

Florianópolis

2018

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Ignácio, Rafael Crispim

Guia Facetado de Técnicas de Elicitação de Requisitos /
Rafael Crispim Ignácio ; orientador, Fabiane Barreto
Vavassori Benitti, 2018.

190 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Sistema de Informação, Florianópolis, 2018.

Inclui referências.

1. Sistema de Informação. 2. Técnicas de elicitação de
requisitos. 3. Elicitação de Requisitos. 4. Engenharia de
Requisitos. 5. Engenharia de Software. I. Benitti, Fabiane
Barreto Vavassori. II. Universidade Federal de Santa
Catarina. Graduação em Sistema de Informação. III. Título.

RAFAEL CRISPIM IGNÁCIO

GUIA FACETADO DE TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

Este Trabalho de conclusão de curso foi julgado aprovado para a obtenção do Título de “ Bacharel em Sistemas de Informação ”, e aprovado em sua forma final pelo curso de Sistemas de Informação.

Florianópolis, 05 de dezembro 2018.

Cristian Koliver
Coordenador do Curso

Banca Examinadora:

Profa. Dra. Fabiane Barreto Vavassori Benitti
Orientadora

Profa. Dra. Patricia Vilain

Prof. Dr. Jean Carlo Rossa Hauck

RESUMO

A engenharia de requisitos é uma das etapas mais importantes de um projeto, impactando direta ou indiretamente no sucesso do mesmo. Durante este processo, existem diversas técnicas que podem ser escolhidas para a elicitação e documentação dos requisitos. No cenário atual, não foram encontradas fontes, sejam elas artigos, pesquisas ou até guias, onde essas técnicas estão devidamente organizadas, facilitando o entendimento das mesmas na identificação suas características. A assertividade na escolha das técnicas pode influenciar na qualidade da fase de elicitação e por consequência no produto desenvolvido. Desta forma, este trabalho tem como objetivo principal levantar as técnicas de elicitação de requisitos existentes, organizando e categorizando de uma forma que auxilie na identificação de quais se adequam ao cenário de determinado projeto. Através do processo de revisão sistemática foram identificadas 96 técnicas de elicitação de requisitos. Para a classificação foram utilizadas 21 facetas distintas também com base no estudo executado. Por fim, no guia desenvolvido, são apresentadas 12 técnicas classificadas por meio de facetas.

Palavras-chave: engenharia de software, elicitação de requisitos, classificação facetada, técnicas de elicitação.

ABSTRACT

Requirements engineering is one of the most important steps in a project, impacting directly or indirectly on its success. During this process, there are several techniques that can be chosen for the elicitation and documentation of requirements. In the current scenario, no sources were found, be they articles, surveys or even guides, where these techniques are properly organized, facilitating their understanding and get their characteristics. The assertiveness in the choice of techniques can influence the quality of the elicitation phase and consequently the developed product. In this way, this work has as main objective to raise the existing elicitation techniques, organizing and categorizing in a way that helps in identifying which ones fit the project scenario. Through the systematic review process, 96 elicitation techniques were identified. For the classification were used 21 distinct facets also based on the study performed. Finally, in the developed guide, 12 techniques classified by facets are presented.

Keywords: software engineering, requirements elicitation, faceted classification, elicitation techniques.

LISTA DE FIGURAS

Figura 1	Fatores de sucesso de um projeto.....	15
Figura 2	Relação entre tipos de requisitos.	23
Figura 3	Divisão dos tipos de requisitos não funcionais.....	24
Figura 4	Exemplo de um ciclo do processo de engenharia de requisitos.....	25
Figura 5	Exemplo de classificação hierárquica.....	27
Figura 6	Exemplo de classificação facetada.	28
Figura 7	Seleção de estudos.....	35
Figura 8	Gráfico de técnicas por frequência nas publicações.....	36
Figura 9	Nuvem de palavras de técnicas conforme a frequência na revisão sistemática	39
Figura 10	Organização do conteúdo apresentado no guia.....	46
Figura 11	Diagrama de casos de uso.....	55
Figura 12	Listar técnicas através do Guia	56
Figura 13	Página de filtros disponíveis	57
Figura 14	Modelo Entidade-Relacionamento.....	58
Figura 15	Comunicação entre camadas da aplicação	59
Figura 16	Mapa de acesso ao guia.....	60
Figura 17	Estrutura hierárquica do modelo GQM.....	61
Figura 18	As quatro fases do método GQM	62
Figura 19	Total de entrevistados por categoria.....	66
Figura 20	Total de entrevistados por experiência	66
Figura 21	Técnicas citadas antes da apresentação do guia.....	69
Figura 22	Técnicas citadas depois da apresentação do guia.....	70
Figura 23	Total de resultados esperados antes da apresentação do guia	70

Figura 24	Total de resultados esperados após da apresentação do guia.....	71
Figura 25	Técnicas citadas que não estão presentes no guia	71
Figura 26	Escalas de avaliação conforme a pontuação SUS	73
Figura 27	Página de login.....	90
Figura 28	Página principal.....	91
Figura 29	Listagem de técnicas	93
Figura 30	Formulário de cadastro de técnicas.....	94
Figura 31	Listagem de facetas	96
Figura 32	Formulário de cadastro de facetas.....	96
Figura 33	Listagem de classificações.....	98
Figura 34	Formulário de cadastro de classificações.....	98
Figura 35	Listagem de usuários	100
Figura 36	Formulário de cadastro de usuários.....	100
Figura 37	Listar técnicas através do Guia	102
Figura 38	Página de filtros disponíveis	103
Figura 39	Visualizar a página de uma técnica	103
Figura 40	Visualizar facetas cadastradas	104
Figura 41	Compartilhamento com redes sociais	105
Figura 42	Diretórios utilizados no desenvolvimento	120

LISTA DE TABELAS

Tabela 1	Fatores de falhas de um projeto.....	15
Tabela 2	Critérios PICOC utilizados na revisão sistemática.....	31
Tabela 3	Técnicas coletadas por publicação.....	37
Tabela 4	Características relacionadas às técnicas de elicitação de requisitos.....	40
Tabela 5	Técnicas tradicionais (ABBASI et al., 2015) (YOUSEF; ALMARABEH, 2015) (SHARMA; PANDEY, 2013) (REHMAN; KHAN; RIAZ, 2013) (SHARMA; PANDEY, 2014)	42
Tabela 6	Técnicas cognitivas (ABBASI et al., 2015) (YOUSEF; ALMARABEH, 2015) (SHARMA; PANDEY, 2013)	42
Tabela 7	Técnicas Inovadoras (SHARMA; PANDEY, 2013)	42
Tabela 8	Técnicas de grupo/colaborativas/criativas (YOUSEF; ALMARABEH, 2015) (YOUSEF; ALMARABEH, 2015) (SHARMA; PANDEY, 2013) (SAHA et al., 2012) (REH- MAN; KHAN; RIAZ, 2013) (DUARTE et al., 2012) (NIJEM, 2013) (GHANBARI; SIMILÄ; MARKKULA, 2015)	43
Tabela 9	Técnicas não categorizadas	43
Tabela 10	Arranjo de facetas	44
Tabela 11	Modelo de organização das facetas	45
Tabela 12	Classificação - Entrevista	49
Tabela 13	Classificação - Prototipação	51
Tabela 14	Objetivo de medição 1 - Avaliação do conteúdo	62
Tabela 15	Questão 1 para o Objetivo 1 e suas respectivas métricas	63
Tabela 16	Caso de exemplo para utilização na coleta de dados	63
Tabela 17	Questão 2 para o Objetivo 1 e suas respectivas métricas	63
Tabela 18	Escala likert utilizada para medições da questão 1.2	63
Tabela 19	Objetivo de medição 2 - Análise da ferramenta	64
Tabela 20	Questão 1 para o Objetivo 2 e suas respectivas métricas	64

Tabela 21	Classificação dos participantes segundo suas informações	65
Tabela 22	Identificação dos entrevistados	67
Tabela 23	Respostas relacionadas às questões acerca do tema proposto	68
Tabela 24	Avaliação do conteúdo produzido e da ferramenta	68
Tabela 25	Publicações selecionadas na revisão sistemática.....	88
Tabela 26	Questões utilizadas na entrevista	108

LISTA DE ABREVIATURAS E SIGLAS

IEEE	Institute of Electrical and Electronic Engineers	14
ISO	International Organization for Standardization	22
IEC	International Electrotechnical Commission	22
CDD	Classificação Decimal de Dewey	26
CDU	Classificação Decimal Universal	26
PME	Personalidade, Matéria e Energia	27
PICOC	Population, Intervention, Comparison, Outcome and Context	31
TCC	Trabalho de conclusão de curso	33
JAD	Joint Application Design	39
FAST	Facilitated Application Specification Techniques	39
CRC	Class Responsibility Collaboration	39
RAD	Rapid Application Development	39
SSM	Soft Systems Methodology	39
EPMcreate	Elementary Pragmatic Model Create	39
POEPMcreate	Power-Only Elementary Pragmatic Model Create	39
CoREA	Collaborative Requirements Elicitation and Analysis	39
QFD	Quality Function Deployment	39
IBIS	Issue based information systems	39
RESCUE	Requirements with Scenarios for a User-centered Environment	39
ARM	Accelerated Requirements Method	39
5W1H	When, Where, Who, What, Why and How	39
RF	Requisitos funcionais	53
RNF	Requisitos não funcionais	53

RN	Requisitos de negócio	53
PHP	PHP: Hypertext Preprocessor	54
HTML	HyperText Markup Language	55
CSS	Cascading Style Sheets	55
UML	Unified Modeling Language	55
MVC	Model, View and Controller	58
API	Application Programming Interface	59
GQM	Goal, Question and Metric	61
SUS	System Usability Scale	64

SUMÁRIO

1 INTRODUÇÃO	14
1.1 PROBLEMAS	16
1.2 SOLUÇÃO	17
1.3 OBJETIVOS	17
1.3.1 Objetivos gerais	17
1.3.2 Objetivos específicos	17
1.4 ESCOPO DO TRABALHO	18
1.5 METODOLOGIA	18
1.5.1 Metodologia de pesquisa	18
1.5.2 Procedimentos metodológicos	19
2 FUNDAMENTAÇÃO BIBLIOGRÁFICA	21
2.1 ENGENHARIA DE REQUISITOS	21
2.1.1 Definição	21
2.1.2 Requisitos	22
2.1.3 Fases de um processo de Engenharia de Requisitos	23
2.1.4 Elicitação de Requisitos	24
2.2 CLASSIFICAÇÃO	25
2.2.1 Classificação Hierárquica	26
2.2.2 Classificação facetada	27
2.2.3 Sistema de classificação	28
3 ESTADO DA ARTE	30
3.1 REVISÃO SISTEMÁTICA	30
3.1.1 Formulação das perguntas	30
3.1.2 Encontrar estudos	31
3.1.3 Avaliar e selecionar estudos	32
3.1.3.1 Critérios de inclusão	33
3.1.3.2 Critérios de exclusão	33
3.1.3.3 Ameaças à validade	33
3.1.4 Resumo e síntese de estudos relevantes	35
3.1.4.1 Q1 - Quais técnicas são utilizadas para elicitação de requisitos?	36
3.1.4.2 Q2 - Quais são as características das técnicas de Elicitação de Requisitos?	39
3.1.5 Determinando a aplicabilidade dos resultados	41

3.1.6 Revisar e avaliar a literatura	41
4 CLASSIFICAÇÃO	42
4.1 COLETA DE DOMÍNIOS	42
4.2 LISTAGEM DE ENTIDADES	43
4.3 CRIAÇÃO DE FACETAS	43
4.4 ARRANJO DE FACETAS	44
4.5 ORDEM DE CITAÇÃO	44
4.6 CLASSIFICAÇÃO	45
4.6.1 Entrevistas	46
4.6.1.1 Prós	47
4.6.1.2 Contras	48
4.6.2 Prototipação	49
4.6.2.1 Prós	50
4.6.2.2 Contras	50
4.7 REVISÃO, TESTE E MANUTENÇÃO	51
4.8 CONSIDERAÇÕES FINAIS	51
5 DESENVOLVIMENTO	53
5.1 REQUISITOS	53
5.1.1 Requisitos funcionais	53
5.1.2 Requisitos não funcionais	54
5.1.3 Requisitos de negócio	55
5.1.4 Casos de uso	55
5.2 UC09 - LISTAR TÉCNICAS	56
5.3 MODELAGEM DE DADOS	57
5.4 APLICAÇÃO	57
5.4.1 Servidor	58
5.4.2 Cliente	59
6 AVALIAÇÃO	61
6.1 PLANEJAMENTO	62
6.2 DEFINIÇÃO	62
6.3 COLETA DE DADOS	65
6.4 INTERPRETAÇÃO	69
7 CONSIDERAÇÕES FINAIS	74
7.1 TRABALHOS FUTUROS	74
REFERÊNCIAS	76

APÊNDICE A – STRINGS DE EXECUÇÃO DA BUSCA.....	84
APÊNDICE B – PUBLICAÇÕES SELECIONADAS	88
APÊNDICE C – CASOS DE USO	90
APÊNDICE D – ARTEFATOS UTILIZADOS NA AVALIAÇÃO	107
APÊNDICE E – Guia Facetado de Técnicas de Elicitação de Requisitos	
- Artigo	110
APÊNDICE F – Desenvolvimento - Código fonte	120

1 INTRODUÇÃO

A importância e a complexidade do desenvolvimento de software vem crescendo cada vez mais com o passar do tempo, segundo Pressman:

Indivíduos, negócios e governos dependem, de forma crescente, de software para decisões estratégicas e táticas, assim como para controle e para operações cotidianas. Se o software falhar, as pessoas e as principais empresas poderão vivenciar desde pequenos inconvenientes a falhas catastróficas. Depreende-se, portanto, que um software deve apresentar qualidade elevada (PRESSMAN, 2011).

A engenharia de software é uma área que contempla diversos assuntos, Sommerville (2011) destaca que "Engenharia de Software é uma abordagem sistemática para a produção de software; ela analisa questões práticas de custo, prazo e confiança, assim como as necessidades dos clientes e produtores de software". O autor ainda cita que "não existem técnicas e métodos universais na engenharia de software adequados a todos os sistemas"(SOMMERVILLE, 2011). Com base nessas questões, surgem novas disciplinas devido à necessidade de especialização em determinados assuntos, como por exemplo, a engenharia de requisitos, no qual tem como um dos aspectos mais importantes, a compreensão das necessidades dos interessados no sucesso do produto.

Uma vez que os produtos de software desempenham um papel vital no suporte a desafios estratégicos e oportunidades nos negócios, é importante que estes produtos funcionem de acordo com os requisitos dos clientes ou dos mercados.

Assim, uma tarefa importante no desenvolvimento de software é a identificação e compreensão dos principais requisitos de negócios para assegurar que os produtos de software sejam totalmente compatíveis e evolutivos com o sistema (WOHLIN et al., 2005).

Oliveira et al. (2012), destaca que em 1993, durante uma conferência do IEEE, "a Engenharia de Requisitos ficou estabelecida como uma disciplina independente". Leite (1994) também defende que "essa atividade é de extrema importância já que a complexidade dos sistemas exige uma maior atenção ao correto entendimento do problema antes do acordo de uma solução final".

Conforme uma pesquisa desenvolvida pelo The Standish Group (1995) e apresentada na **Tabela 1**, a engenharia de requisitos está envolvida diretamente com 25.1% (itens 1 e 2) dos problemas encontrados em software. Ainda cabe destacar que 27.3% (itens 3,7, 8, 9) dos fatores podem estar indiretamente vinculados a mesma questão.

Os fatores apresentados na **Tabela 1** sofreram reformulações com o passar tempo. Atualmente o relatório apresenta uma tabela associada aos fatores de sucesso de um

Tabela 1 – Fatores de falhas de um projeto

Fatores de falhas		% Respostas
1.	Falta de entradas de usuário	12.8%
2.	Requisitos e especificações incompletas	12.3%
3.	Mudança de requisitos	11.8%
4.	Falta de apoio executivo	7.5%
5.	Tecnologia imatura	7.0%
6.	Falta de recursos	6.4%
7.	Expectativas irreais	5.9%
8.	Objetivos obscuros	5.3%
9.	Tempo/cronogramas irreais	4.3%
10.	Novas tecnologias	3.7%
11.	Outros	23.0%

Fonte: (THE STANDISH GROUP, 1995)

projeto, conforme apresentado na **Figura 1**. Estes fatores atualizados trazem as características associadas ao sucesso de um projeto e o quanto é investido para cada elemento. Cabe destacar que dois dos itens, envolvimento do usuário e clareza nos objetivos de negócio, podem ser associados à engenharia de requisitos.

Figura 1 – Fatores de sucesso de um projeto.

FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

Fonte: (THE STANDISH GROUP, 2015)

Segundo Sommerville (2007), "O objetivo do processo de engenharia de requisitos é criar e manter um documento de requisitos de sistema". O autor ainda destaca que, em geral, existem quatro subprocessos de alto nível relacionados a esta engenharia:

- Estudo de viabilidade;
- Elicitação e análise de requisitos;
- Validação de requisitos;
- Gerenciamento de requisitos;

Para elicitação de requisitos, o autor destaca que:

Nessa atividade, os engenheiros de software trabalham com os clientes e os usuários finais do sistema para aprender sobre o domínio da aplicação, quais serviços o sistema deve fornecer, o desempenho esperado do sistema, restrições de hardware etc (SOMMERVILLE, 2007, p. 97).

Para Kuchmistaya (2001), a elicitação de requisitos "é uma das partes mais importantes do processo, que pode levar ao sucesso ou ao fracasso do projeto". Segundo Wiegers e Beatty (2013), "em nenhum outro momento do projeto do sistema os interesses dos usuários e clientes cruzam-se mais do que no processo de elicitação de requisitos. Quando bem feita, essa intersecção pode produzir softwares bem elaborados".

Este trabalho procura fazer uma contribuição na área de Engenharia de Software, especificamente em Engenharia de Requisitos, através do levantamento e análise de técnicas de elicitação de requisitos. Ao final será disponibilizado um guia facetado na forma de uma aplicação web, almejando facilitar a identificação das técnicas através de suas características. Cabe ressaltar que, conforme exposto por Tzitzikas e Analyti (2007), uma faceta representa um conjunto de informações associadas e sintetizadas em um nível mais alto de classificação conceitual, possibilitando a organização das técnicas de elicitação de requisitos através de suas características.

1.1 PROBLEMAS

Segundo Wohlin et al. (2005), engenharia de requisitos refere-se a todas as atividades do ciclo de vida relacionadas aos requisitos. Isso inclui principalmente reunir, documentar e gerenciar requisitos. Para cada uma destas etapas, existem técnicas que visam a obtenção de um resultado mais assertivo conforme suas necessidades. Leite (2008), acerca desta informação aponta que, "tendo em vista que o sucesso de um projeto de software será medido a partir da satisfação de seus usuários e partes interessadas no mesmo, faz-se necessário ter uma abordagem disciplinada para definir estas necessidades".

Com relação a atividade de elicitação de requisitos, o autor informa que ela "pode ser feita de diversas formas e diversos fatores podem influenciar a mesma, notoriamente,

questões relacionadas ao ambiente no qual a elicitação de requisitos é feita e a finalidade para qual o sistema se destina”(LEITE, 2008). Como estas formas dependem de fatores externos e internos relacionados ao projeto, a decisão sobre quais técnicas aplicar pode se tornar um problema.

1.2 SOLUÇÃO

Com base no problema relatado, este estudo pretende, através da definição de facetas, caracterizar as principais técnicas de elicitação de requisitos. Conforme descrito por Belgamo, Eduardo e Martins (2008), através da análise das técnicas existentes, é possível mapeá-las através de características comuns. O conteúdo resultante possibilita uma decisão pautada nos aspectos do projeto em questão. Ao final, uma ferramenta de busca será desenvolvida tendo com base as técnicas levantadas e suas respectivas facetas. Seu principal objetivo é servir de interface para que um usuário possa, através dos fatores internos e externos do projeto, descobrir quais técnicas de elicitação de requisitos se enquadram melhor em sua situação. A coleta das informações será feita através de uma revisão sistemática, cujo processo e todos seus critérios de seleção de estudos estão detalhados no **Capítulo 4**.

1.3 OBJETIVOS

Esta seção busca formalizar os objetivos do presente trabalho.

1.3.1 Objetivos gerais

Desenvolver uma ferramenta de busca orientada a facetas para suporte a identificação de técnicas de elicitação de requisitos.

1.3.2 Objetivos específicos

- Levantamento das técnicas de elicitação de requisitos existentes na literatura;
- Propor as facetas e aplica-las às técnicas de elicitação de requisitos;
- Desenvolvimento de uma aplicação web para busca de técnicas de elicitação com base em um sistema de classificação;

1.4 ESCOPO DO TRABALHO

Este trabalho se restringe a compreender e organizar somente as técnicas de elicitação de requisitos. Conforme citado por Batista e Carvalho (2003), "Existem diversas técnicas de elicitar requisitos, tanto a partir de pessoas como a partir de documentos ou observações". Compreende-se então, que devido a quantidade e diversidade existentes, as técnicas de elicitação de requisitos necessitam da devida atenção para sua classificação e organização. Desta forma, neste trabalho não serão avaliadas as técnicas de documentação nem de qualquer outro subprocesso da engenharia de requisitos. Entende-se que as técnicas relacionadas aos outros subprocessos desta engenharia necessitam de uma abordagem de igual importância para sua plena compreensão.

As técnicas de elicitação de requisitos serão pesquisadas conforme os parâmetros definidos na pesquisa sistemática e as facetas propostas serão definidas conforme o levantamento de dados destas técnicas. Estas facetas delimitarão a ferramenta de busca, e por consequência seu espectro de atuação. Caso for encontrado um número alto de técnicas de elicitação de requisitos, inviabilizando a execução da classificação, será utilizada uma amostra composta pelas técnicas mais frequentes nas publicações selecionadas.

1.5 METODOLOGIA

A fim de descrever os métodos adotados no presente trabalho, esta seção apresenta dois tópicos. O primeiro define a metodologia de pesquisa e o segundo faz menção ao procedimento metodológico.

1.5.1 Metodologia de pesquisa

Para atingir os objetivos deste trabalho, será utilizada a metodologia de pesquisa indutiva, segundo Gil (2008), neste método "parte-se da observação de fatos ou fenômenos cujas causas se deseja conhecer. A seguir, procura-se compará-los com a finalidade de descobrir as relações existentes entre eles. Por fim procede-se à generalização, com base na relação verificada entre os fatos e os fenômenos". Desta forma, a utilização do método indutivo neste trabalho tem como objetivo identificar a relação das técnicas de elicitação através de suas características, insumos necessários para aplicação e impactos no meio onde são aplicados.

Partindo do ponto de vista dos objetivos, o presente trabalho trata-se de uma pesquisa exploratória, para Gil (2008), "pesquisas exploratórias são desenvolvidas com o

objetivo de proporcionar visão geral, de tipo aproximativo, acerca de determinado fato”. Através de uma pesquisa sistematizada, será possível avaliar um universo controlado de técnicas de elicitação de requisitos que possuem características similares.

Do ponto de vista de sua natureza, o trabalho consiste em uma pesquisa aplicada, gerando insumos que podem vir a ser utilizados na escolha da técnica de elicitação. Para Silva e Menezes (2001), uma pesquisa aplicada “objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos”.

Através de critérios preestabelecidos, serão levantadas informações sobre as técnicas de elicitação de requisitos. Os resultados coletados serão classificados de forma qualitativa. Quanto ao procedimento técnico, o trabalho vai utilizar o método de pesquisa bibliográfica, “desenvolvida a partir de material já elaborado, constituído principalmente de livros e artigos científicos”(BATISTA; CARVALHO, 2003).

1.5.2 Procedimentos metodológicos

Este trabalho será composto pelas seguintes etapas:

1. Gerência do projeto: Esta etapa visa a análise da viabilidade do projeto bem como todo seu planejamento
 - (a) Planejamento: Visa organizar o plano de execução do trabalho proposto
 - (b) Monitoramento e controle: Possui o objetivo prover métodos de garantir que o trabalho ocorre dentro do plano previsto
2. Fundamentação bibliográfica: Esta etapa tem a função de fornecer embasamento para os conteúdos relacionados ao tema da pesquisa
 - (a) Engenharia de requisitos: levantamento da bibliografia acerca do tema proposto no trabalho
 - (b) Taxonomia e esquemas de classificação: Visa levantar e compreender os métodos de classificação através de facetas
3. Estado da arte: Esta etapa tem o objetivo de identificar e analisar as técnicas de elicitação de requisitos existentes na bibliografia. Conforme proposto por Kitchenham (2010), será realizada uma revisão sistemática almejando uma visão do estado da arte do tema dentro de sua linha de pesquisa através das seguintes atividades:
 - (a) Desenvolvimento do protocolo de busca: Especificação dos objetivos, critérios para inclusão e exclusão dos estudos.

- (b) Execução da busca: Identificação, seleção e avaliação dos estudos.
 - (c) Análise dos resultados: Análise e normalização dos dados captados, realização de uma síntese e descrição das evidências.
4. Taxonomia de classificação: Compreende a proposta e aplicação das facetas aos dados levantados
- (a) Definição das facetas: Especificação da classificação das facetas
 - (b) Execução: Aplicar a classificação nos dados levantados
 - (c) Análise dos resultados: Analisar e realizar uma síntese dos resultados obtidos
5. Desenvolvimento da aplicação: Compreende o desenvolvimento da aplicação de busca onde serão disponibilizados os resultados obtidos com os objetivos 1 e 2
- (a) Planejamento: Levantar os requisitos e definir a especificação da aplicação de busca
 - (b) Desenvolvimento: Desenvolver aplicação
 - (c) Testes: Analisar os resultados da aplicação
6. Avaliação: Avaliar os resultados obtidos junto aos 3 objetivos específicos através de uma pesquisa participante ou pesquisa-ação. Esta fase tem o objetivo de verificar se a ferramenta proposta está adequada para utilização dos usuários e também se as facetas estão adequadas a realidade da área de engenharia de requisitos
- (a) Planejamento: Definição do processo a ser realizado para avaliação dos objetivos expostos
 - (b) Execução: Executar o processo de avaliação definido
 - (c) Análise dos resultados: Sintetizar os resultados da avaliação

2 FUNDAMENTAÇÃO BIBLIOGRÁFICA

Esta seção fornece embasamento para os assuntos relacionado a área de pesquisa. Inicialmente são apresentados os conceitos principais envolvendo Engenharia de Requisitos, em seguida é apresentada a Elicitação de Requisitos, assunto no qual a pesquisa é direcionada. Por fim, são apresentados os conceitos em torno das taxonomias e processos de classificação que são abordados durante a pesquisa.

2.1 ENGENHARIA DE REQUISITOS

Conforme o desenvolvimento de software foi evoluindo, novas disciplinas foram surgindo com o objetivo de contemplar as necessidades apresentadas. A Engenharia de Requisitos nasceu de uma dessas necessidades, a partir da Engenharia de Software, e hoje é considerada como um dos itens mais importantes da área. Para Leite (1994), essa atividade “é de extrema importância já que a complexidade dos sistemas exige uma maior atenção ao correto entendimento do problema antes do acordo de uma solução final”. Wohlin et al. (2005) também citam que a Engenharia de Requisitos é “uma das etapas cruciais para o design e desenvolvimento de software, pois aborda problemas críticos de projetar o software certo para o cliente”.

2.1.1 Definição

O termo Engenharia de Requisitos é definido pela ISO/IEC/IEEE International Standard (2011) como um processo interdisciplinar entre os domínios do cliente e do fornecedor para estabelecer e manter os requisitos a serem atendidos pelo sistema, software ou serviço de interesse. Desta maneira, cabe destacar que esta disciplina não é composta apenas com conhecimentos técnicos em software, mas também com conhecimentos relacionados ao problema do cliente.

Leite (1994) define Engenharia de Requisitos como uma “disciplina que procura sistematizar o processo de definição de requisitos”. Segundo ele, esta “sistematização é necessária porque a complexidade dos sistemas exige que se preste mais atenção ao correto entendimento do problema antes do comprometimento de uma solução”.

Outros autores também seguem o mesmo raciocínio, Sommerville (2011) por exemplo, destaca que Engenharia de Requisitos é “o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao

desenvolvimento do sistema”. O autor ainda defende que “é um estágio particularmente crítico do processo de software, pois erros nessa fase inevitavelmente geram problemas no projeto e na implementação do sistema”. Para Wohlin et al. (2005), essa disciplina é reconhecida como uma base para as funcionalidades do sistema, sendo um fator determinante na qualidade do software. Wiegers e Beatty (2013) ainda citam que as entradas inadequadas dos usuários e as deficiências quanto ao gerenciamento dos requisitos dos clientes são fatores importantes para projetos mal sucedidos, mas apesar dessas evidências as empresas continuam praticando métodos ineficazes em relação aos requisitos.

2.1.2 Requisitos

Conforme destaca Sommerville:

O termo requisito não é usado de forma consistente pela indústria de software. Em alguns casos, o requisito é apenas uma declaração abstrata em alto nível de um serviço que o sistema deve oferecer ou uma restrição a um sistema. No outro extremo, é uma definição detalhada e formal de uma função do sistema. (SOMMERVILLE, 2011)

De acordo com a ISO/IEC/IEEE International Standard (2011), um requisito é uma declaração que expressa uma necessidade ou suas restrições e condições associadas, podendo existir diferentes níveis. Exemplificando a definição da IEEE citada anteriormente, Wiegers e Beatty (2013) representam através de uma imagem (**Figura 2**), uma separação que leva em consideração os níveis de aplicação e tipos de requisitos.

Para Sommerville, a separação ocorre da seguinte forma:

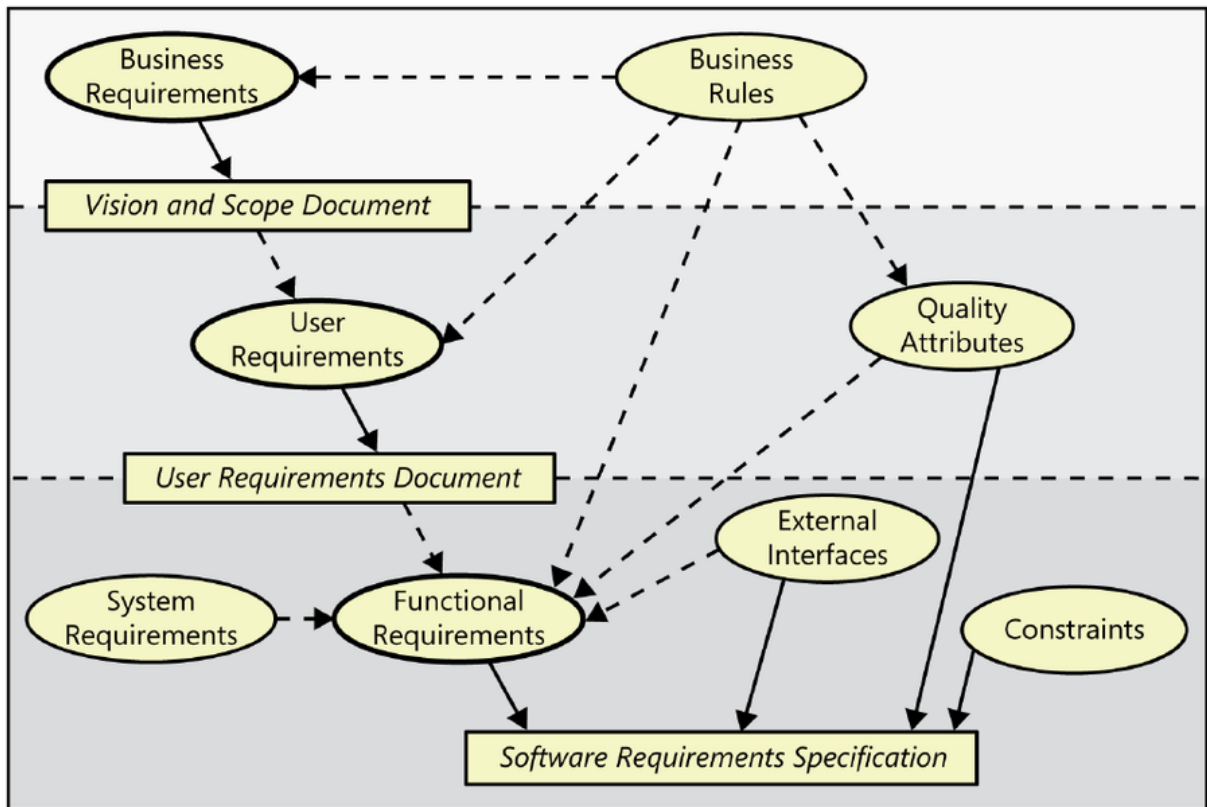
- Requisitos funcionais

São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer. (SOMMERVILLE, 2011)

- Requisitos não funcionais

São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo. (SOMMERVILLE, 2011)

Figura 2 – Relação entre tipos de requisitos.



Fonte: (WIEGERS; BEATTY, 2013)

O autor ainda destaca que os requisitos não funcionais podem se subdividir em diversos tipos (**Figura 3**).

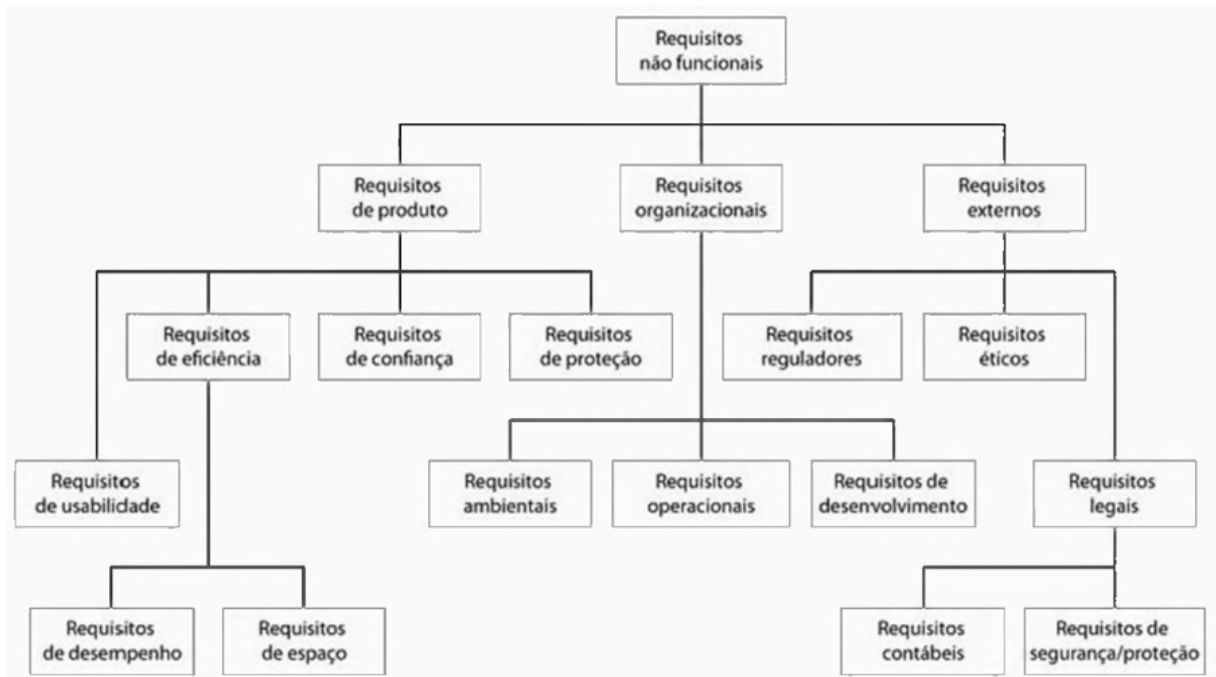
2.1.3 Fases de um processo de Engenharia de Requisitos

O processo de Engenharia de Requisitos, de um modo geral, é dividido em fases, mas não existe um consenso entre os autores quanto a divisão ou nomenclatura das mesmas. Jarke e Pohl (1994) destaca que o ciclo da Engenharia de Requisitos clássica consiste em três fases: Elicitação, Documentação e Validação.

Em contraponto, outros autores indicam uma divisão de mais fases para o processo de ER. Pressman (2011), por exemplo, separa o processo em: Concepção, Levantamento, Elaboração, Negociação, Especificação e Validação e Gestão de Requisitos. Já Sommerville (2011) defende que o processo é composto pelas fases: Estudo de viabilidade, Elicitação e análise de requisitos, Especificação de requisitos, Validação e Gerenciamento de requisitos. Wohlin et al. (2005) dividem as fases em: Elicitação, Análise e Documentação, Negociação, Verificação e Validação, Gerência de mudanças e Rastreamento de requisitos.

Desta forma, é importante destacar que o presente trabalho utilizará a terminologia

Figura 3 – Divisão dos tipos de requisitos não funcionais.



Fonte: (SOMMERVILLE, 2011)

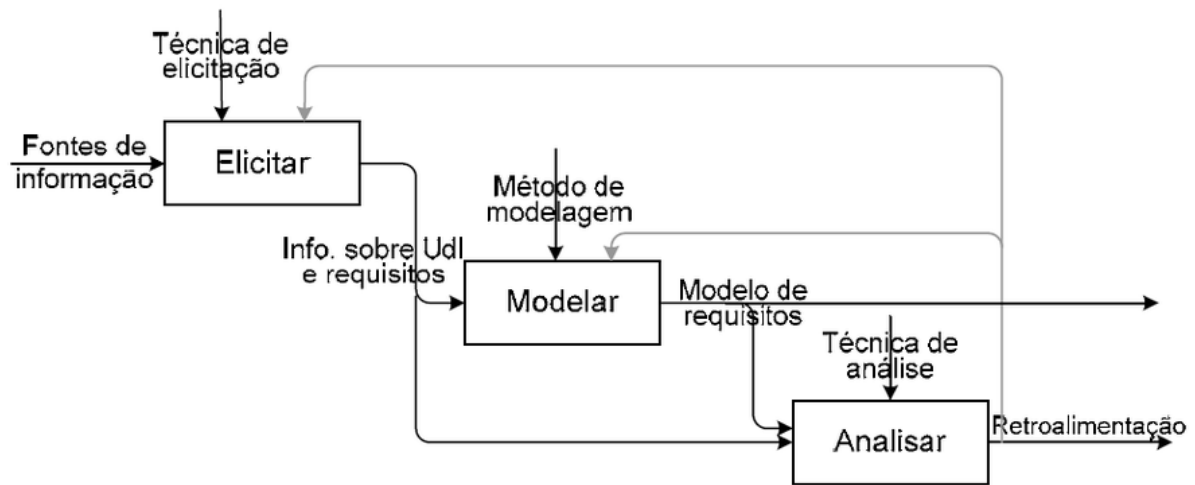
Elicitação de requisitos quando se referir a esta fase do processo. Esta será a fase central deste projeto, visto que esta pesquisa tem o objetivo de analisar as técnicas de elicitação de requisitos. Apesar de Pressman não utilizar essa terminologia, o autor também contempla esta fase referenciando a ela como levantamento de requisitos. Leite (1988) demonstra graficamente como um ciclo de processo de engenharia de requisitos pode ocorrer. A elicitação de requisitos se destaca por ser a fase inicial do ciclo, fornecendo a compreensão dos requisitos, conforme representado na **Figura 4**.

2.1.4 Elicitação de Requisitos

O coração do desenvolvimento de requisitos é a elicitação, o processo de identificação das necessidades e restrições das várias partes interessadas para um sistema de software (WIEGERS; BEATTY, 2013). Nesta fase, “os engenheiros de software trabalham com clientes e usuários finais do sistema para obter informações sobre o domínio da aplicação, os serviços que o sistema deve oferecer, o desempenho do sistema, restrições de hardware e assim por diante”(SOMMERVILLE, 2011).

De acordo com a definição da ISO/IEC/IEEE International Standard (2011), a Elicitação de Requisitos é uma fase da Engenharia de Requisitos através da qual o adquirente e os fornecedores de um sistema descobrem, analisam, articulam, compreendem e docu-

Figura 4 – Exemplo de um ciclo do processo de engenharia de requisitos.



Fonte: (LEITE, 1988)

mentam os requisitos do sistema e os processos do ciclo de vida. Alguns autores ainda utilizam outra nomenclatura para esta fase. Pressman (2011), por exemplo, utiliza a nomenclatura levantamento, conforme dito na seção anterior.

Segundo Soares e Moura (2015), uma forma de reduzir os problemas do uso de linguagem natural é a definição de modelos e padrões que ajudam os engenheiros de requisitos e usuários a elicitar, expressar e registrar informações relevantes no domínio do sistema. Batista e Carvalho (2003) destaca que “diversas técnicas podem ser aplicadas e, cada vez mais, as tradicionais estão convivendo com novas técnicas, com o objetivo de aprimorar a identificação dos requisitos e diminuir os problemas decorrentes de uma má elicitação”. Para Davis et al. (2006), “uma variedade de métodos ou técnicas foram propostas para adquirir informações para elicitação, tais como: entrevista, análise de protocolo, grade de repertório, grupos de trabalho, etc.”. Além disso, o autor ressalta que estas técnicas podem variar ou serem combinadas.

2.2 CLASSIFICAÇÃO

Uma das definições que mais se adequam ao contexto deste trabalho foi apresentada por Piedade (1977). Nesta definição a autora destaca que classificar é, “dividir em grupos ou classes, segundo as diferenças e semelhanças. É dispor os conceitos segundo suas semelhanças e diferença em certo número de grupos metodicamente distribuídos”(PIEADADE, 1977). Ainda com relação a esta percepção, Nunes et al. (2007) destaca que classificação pode ser entendido como “a ordenação de um conjunto de seres em pequenos agrupamentos, de acordo com características que os unem ou diferem de outros grupos.”.

Para Nunes et al. (2007), “as leis de classificação devem fundamentar-se nesse princípio, para que sejam legítimas”. Eles ressaltam que:

A classificação é uma função importante para a transparência e para o compartilhamento de informações, as quais são caminhos para tomadas de decisões, para a preservação da memória técnica e administrativa das organizações contemporâneas e também para o exercício da cidadania. (NUNES et al., 2007)

Como destacado por Silva:

Muitos foram os sistemas de classificação desenvolvidos ao longo da história humana, passando pelas filosóficas, depois bibliográficas até os esquemas de organização do conhecimento considerados mais modernos como Tesouros e Ontologias. Os sistemas de classificação dependem destes termos. Como exemplos de sistemas, citemos os mais usados no mundo, que são a Classificação Decimal de Dewey (CDD) e a Classificação Decimal Universal (CDU). (SILVA et al., 2011)

De maneira geral, as classificações são organizadas em dois grupos, a classificação hierárquica e a classificação facetada.

2.2.1 Classificação Hierárquica

Esse tipo de classificação possui este nome devido a forma que organiza os conceitos, ela teve suas primeiras contribuições na lógica aristotélica (LANGRIDGE, 1977). Araújo (2006) destaca que as noções de classificação que surgiram desde sua primeira concepção, “foram fundamentais na formulação dos primeiros sistemas de classificação bibliográfica”.

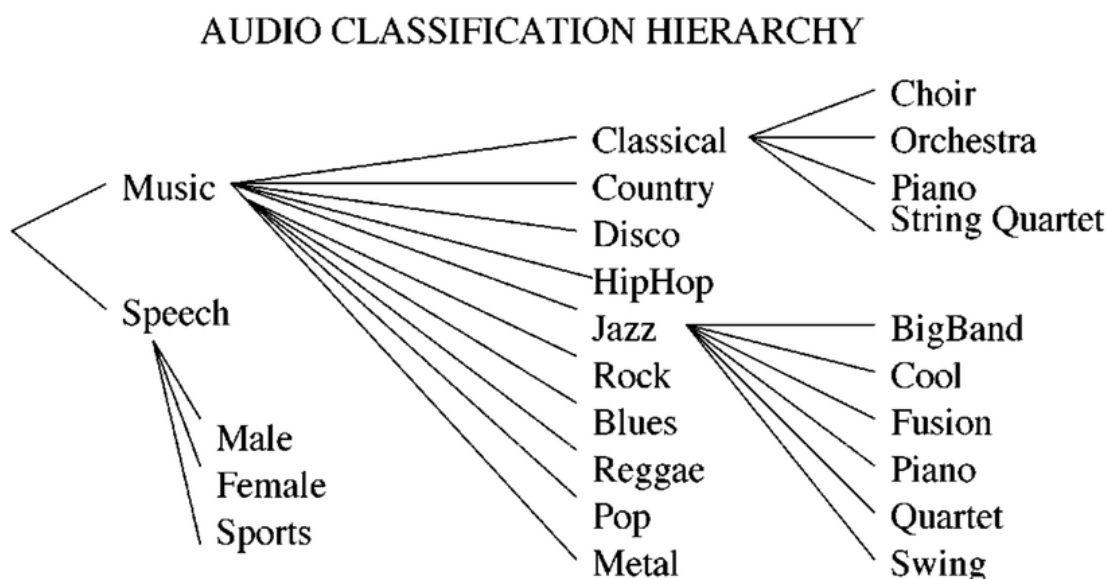
Neste tipo de classificação:

Os documentos podem ser ordenados utilizando vários critérios, como o tamanho e a cor do livro, mas nas classificações, os documentos são ordenados pelo assunto que cada um apresenta, assim facilita a localização das informações em comum nas estantes ou nos arquivos. Essas são as duas classificações mais utilizadas atualmente, são a Classificação Decimal de Dewey (CDD) e a Classificação Decimal Universal (CDU), que classificam os documentos pelo assunto. (EDUVIRGES, 2011)

A **Figura 5**, traz um exemplo de classificação hierárquica.

Apesar de ser uma estrutura hierárquica de fácil utilização, este tipo de organização não foi considerado adequado para o tema apresentado neste trabalho. Batista e Carvalho (2003) diz que a estrutura deste sistema “é fácil de compreender e usar, porém não é flexível”, tornando inviável a utilização para classificação das técnicas de elicitação de requisitos.

Figura 5 – Exemplo de classificação hierárquica.



Fonte: (TZITZIKAS; ANALYTI, 2007)

2.2.2 Classificação facetada

Para Tristão, Fachin e Alarcon (2004), “classificação facetada é conhecida como um esquema analítico sintético porque envolve dois processos distintos: a análise do assunto em facetas e a síntese dos elementos que constituem o mesmo, sendo, portanto, aplicável a qualquer área do conhecimento.”

Desenvolvida por Shiyali Ramamrita Ranganathan na década de 30, a partir da Colon Classification, tabela de classificação elaborada para a organização do acervo da Biblioteca da Universidade de Madras, na Índia [...] as tabelas existentes não apresentavam as bases teóricas para sua elaboração. Ranganathan foi o primeiro a evidenciar os princípios utilizados na elaboração de sua tabela, proporcionando uma verdadeira revolução na área da Classificação Bibliográfica (CAMPOS, 2001)

Mills (2004) destaca que este é o tipo de classificação mais usado nos últimos cinquenta anos em bibliotecas e tem como característica fundamental a divisão de qualquer assunto de acordo com as regras de divisão lógica.

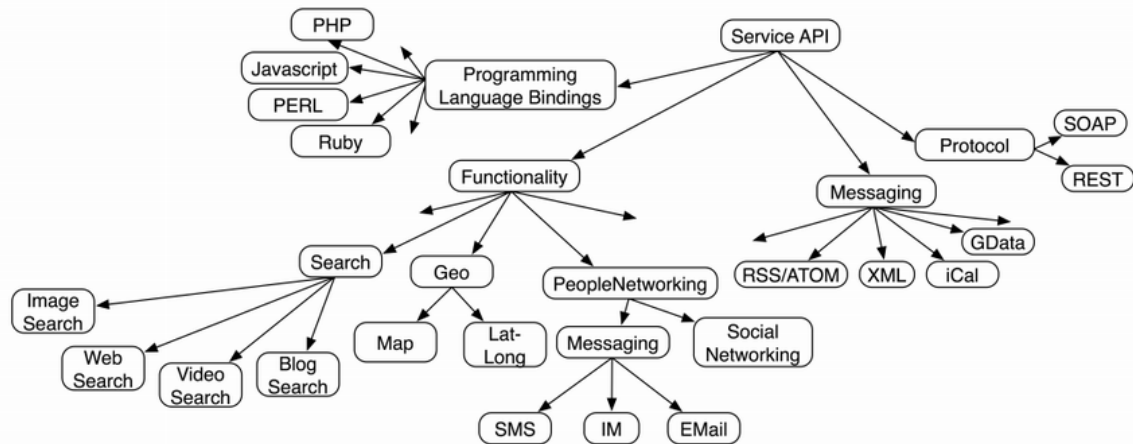
A primeira classificação facetada foi criada por Ranganathan:

Na sua Colon Classification, determinou apenas 5 facetas a que chamou de: P = Personalidade; M = Matéria; E = Energia; S = Space (local); T = Time (época). Mais tarde verificando que elas não eram suficientes para classificar assuntos muito complexos, criou os ciclos (rounds) e níveis (levels). Isto significa que as facetas PME podem aparecer mais de uma vez em certos assuntos, desde que o primeiro ciclo tenha se

completado, o que só é possível depois que a faceta Energia aparece. (BARBOSA, 1972)

Uma classificação facetada pode ser exemplificada conforme apresentado na **Figura 6**.

Figura 6 – Exemplo de classificação facetada.



Fonte: (RANABAHU et al., 2008)

As principais características que levam à escolha da classificação facetada ao invés de uma classificação hierárquica são:

- A classificação facetada objetiva a estruturação do conhecimento, através da organização de seus conceitos e da criação de relacionamento entre eles (SILVA et al., 2011)
- Permite o mapeamento de uma área de assunto e a inclusão de novos conceitos, sem alterar a estrutura do sistema (SILVA et al., 2011)
- Não dependem da segmentação de um universo (BATISTA; CARVALHO, 2003)

2.2.3 Sistema de classificação

Com o passar do tempo, novos sistemas de classificação foram criados e com eles a metodologia vem sendo aprimorada. Neste trabalho, devido a sua simplicidade e completez, optou-se em utilizar uma metodologia apresentada por Denton (2003) que consiste em um procedimento com os sete passos:

1. Coleta de Domínios: Coletar uma amostra representativa das entidades

2. Listagem de Entidades: Listar as entidades organizando os conceitos
3. Criação de Facetas: Examinar os termos resultantes, examinando e reduzindo em facetas mutuamente exclusivas e conjuntamente exaustivas
4. Arranjo de Facetas: Esta etapa contém duas tarefas. Primeiro deve ser feita uma ordem de teste de todos termos sob as facetas. Por fim, após testar e validar o rascunho das facetas, é necessário fazer um arranjo final
5. Ordem de Citação: Esta fase faz parte do plano notacional. Diz respeito a qual ordem de citação padrão que será a maneira padrão como as coisas serão ordenadas
6. Classificação: Esta é a fase de execução onde o domínio será classificado conforme as facetas definidas
7. Revisão, teste e manutenção: Se for detectado algum problema na sexta etapa, deve-se voltar quantas etapas forem necessárias para ajustar a classificação

3 ESTADO DA ARTE

Esta seção tem o objetivo de identificar e analisar as técnicas de elicitação de requisitos existentes na bibliografia por meio de uma revisão sistemática. Segundo Kitchenham (2010), uma revisão sistemática da literatura é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma determinada questão de pesquisa, área de tópico ou fenômeno de interesse. Existem alguns motivos para se desempenhar uma revisão sistemática, Kitchenham destaca como mais comuns os seguintes:

- Para resumir a evidência existente sobre um tratamento ou tecnologia, e para resumir a evidência empírica dos benefícios e limitações de um método específico;
 - Identificar lacunas na pesquisa atual, a fim de sugerir áreas para uma investigação mais aprofundada;
 - Fornecer uma estrutura / fundo para posicionar adequadamente novas atividades de pesquisa;
- (KITCHENHAM, 2010)

3.1 REVISÃO SISTEMÁTICA

Uma revisão sistemática é composta de algumas etapas:

- Formulação de perguntas;
- Encontrar estudos;
- Avaliar e selecionar os estudos;
- Resumo e síntese de estudos relevantes;
- Determinando a aplicabilidade dos resultados; e
- Revisar e avaliar a literatura

(NATIONAL HEALTH AND MEDICAL RESEARCH COUNCIL, 2000)

3.1.1 Formulação das perguntas

Kitchenham (2010) destaca que, conforme os guias médicos, a estrutura de uma questão se divide principalmente em três pontos: população, intervenção e resultados.

Petticrew e Roberts (2008) sugerem a utilização dos critérios PICOC (população, intervenção, comparação, resultados e contexto), ampliando as diretrizes anteriores.

Utilizando a estrutura de revisão sistemática destacada pelo National Health and Medical Research Council (2000), foi dado início ao processo de revisão levantando as seguintes questões que pretende-se responder com o presente trabalho:

Q1. Quais técnicas são utilizadas para elicitação de requisitos?

Q2. Quais são as características das técnicas de Elicitação de Requisitos?

Os critérios da pesquisa são destacados na **Tabela 2**.

Tabela 2 – Critérios PICOC utilizados na revisão sistemática

Critério	Descrição	Termos levantados
População	A revisão deve considerar os artigos relacionados a área de requisitos de software com data de publicação de 01/01/2008 até 31/12/2017 (últimos 10 anos).	"software requirements"
Intervenção	A partir da área de requisitos de software, o objetivo é estudar somente as técnicas de elicitação de requisitos e seus sinônimos.	"elicitation techniques"OR "acquisition techniques"OR "elicitation methods"OR "requirements elicitation"
Comparação	Como critério de comparação foram levantados termos que levam a artigos que procuram agrupar técnicas de elicitação de requisitos.	"comparison"OR "comparative study"
Resultados	Assim como no critério de comparação, no critério de resultados são esperados artigos que descrevem um conjunto de técnicas de elicitação de requisitos.	"evaluation"OR "survey"OR "review"OR "mapping"OR "categories"OR "taxonomies"OR "classifications"

Fonte: Elaboração do autor, 2018

3.1.2 Encontrar estudos

Conforme citado por Brereton et al. (2007), a execução da busca deve considerar as seguintes bases:

- IEEEExplore;
- ACM Digital Library;
- Google scholar;
- Citeseer library;
- ScienceDirect;

Com o intuito de complementar os resultados, outras bases, que também tem relação com o tema proposto, devem ser utilizadas:

- Springer;
- Scopus;
- Engineering Village;

Cada uma destas bases científicas possui uma forma específica de execução das buscas, mas em todos os casos os termos estabelecidos na **Seção 3.1.1** são considerados. Não existe uma ordem de prioridade entre as bases utilizadas, apenas é necessário que os artigos sigam os critérios definidos no mapeamento. A junção dos termos forma a seguinte string de busca:

```
("software requirements") AND
("elicitation techniques" OR "acquisition techniques" OR
 "elicitation methods" OR "requirements elicitation") AND
("evaluation" OR "comparison" OR "survey" OR
 "comparative study" OR "review" OR "mapping" OR
 "categories" OR "taxonomies" OR "classifications")
```

As strings utilizadas especificamente para cada base estão detalhadas no **Apêndice A**.

3.1.3 Avaliar e selecionar estudos

A seleção dos estudos é o processo no qual os estudos obtidos através das bases científicas são escolhidos para fazer parte do trabalho. Kitchenham (2007) informa que “seleção de estudos é um processo que ocorre em multiestágios [...] interpretando os critérios de seleção inicialmente no título e no resumo, se o trabalho em questão não for excluído, deve ser avaliado por completo”. Os critérios de seleção tem como objetivo, “identificar os estudos que fornecem evidências diretas sobre a questão de pesquisa”(KITCHENHAM, 2007).

3.1.3.1 Critérios de inclusão

Para o presente trabalho foram levantados os seguintes critérios de inclusão:

- Somente publicações com revisão por pares, como por exemplo conferências e journals;
- Deve contemplar 3 ou mais técnicas de elicitação, podendo ou não apresentar características das mesma;
- Deve ter sido publicado nos últimos 10 anos, de 01/01/2008 até 31/12/2017;

3.1.3.2 Critérios de exclusão

Da mesma forma foram elencados os critérios de exclusão, ou seja, características que definem que o estudo não deve ser selecionado:

- Publicações que não são revisadas por pares, como por exemplo, livros, teses, TCC's, etc;
- Publicações como resumos, posters, tutoriais, serão excluídos;
- Estudos publicados em um idioma diferente do inglês;
- Artigos que não são acessíveis publicamente ou com autorização da universidade;

Além destes critérios, cabe ressaltar que se caso dois artigos apresentam o mesmo resultado, apenas o mais recente será considerado.

3.1.3.3 Ameaças à validade

Segundo Zhou et al. (2016), “a validade é uma propriedade de inferência das revisões sistemáticas e essas pesquisas possuem uma variedade de ameaças à esta característica”. Resumidamente “as ameaças à validade são influências que podem limitar a interpretação das conclusões provenientes dos dados extraídos”(SOUZA; FANTINATO, 2013). As ameaças foram analisadas segundo as taxonomias: validade de construção, validade interna, validade externa e validade de conclusão.

Validade de construção: Visando responder as questões levantadas, foi elaborada uma *string* de busca com o objetivo de encontrar estudos secundários acerca do tema

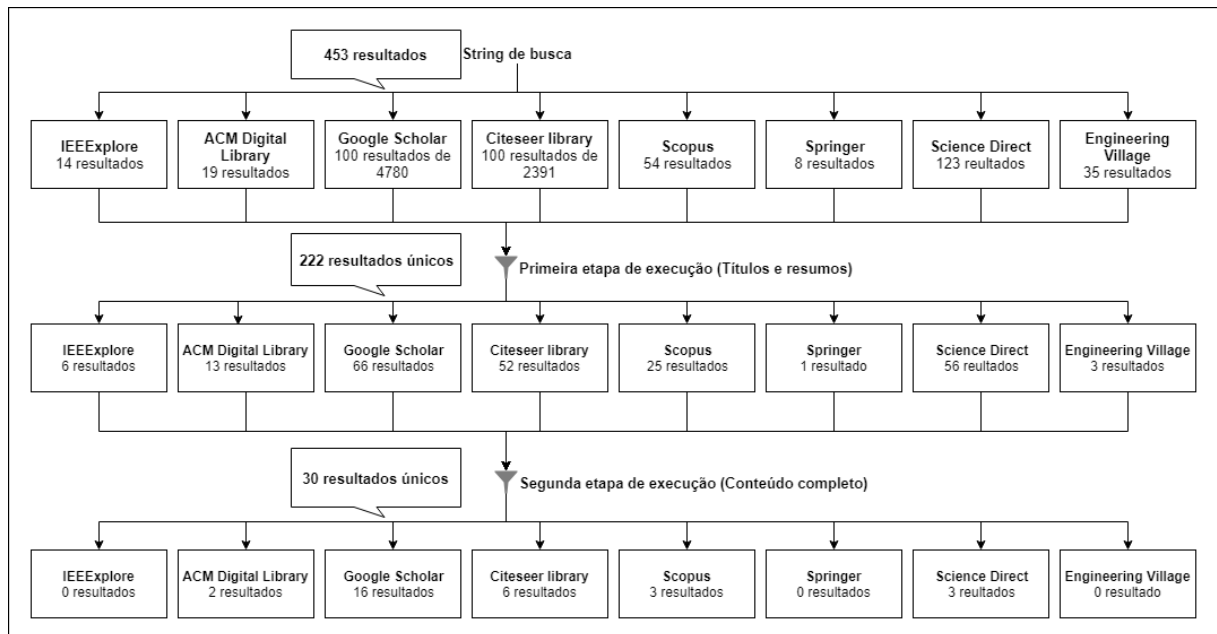
do trabalho. Foram considerados quaisquer tipos de estudos, sejam eles primários, secundários ou terciários, os fatores determinantes para sua utilização foram os critérios de inclusão e exclusão definidos nas **Seções 3.1.3.1 e 3.1.3.2**. Foram utilizadas bases científicas especializadas no tema e genéricas, podendo gerar um resultado muito amplo. Este fator é minimizado devido a especificidade dos termos usados na *string* de busca e também considerando os critérios de inclusão e exclusão associados à revisão sistemática. Apesar da quantidade de bases selecionadas ser alta, ainda é possível que uma base científica importante não tenha sido selecionada. Para as bases que agregam resultados de diversas fontes, como por exemplo o *Google Scholar* e o *Citeseer*, foi definida uma quantidade limite de estudos utilizados. Este fator pode ter um impacto negativo, excluindo um estudo que deveria ser considerado. Para minimizar esse fator foi utilizada a ordem de relevância de cada plataforma como critério.

Validade interna: Os trabalhos selecionados passaram por dois processos. O primeiro mais generalista, analisando os títulos e *abstract* e verificando a adequação com os critérios definidos no trabalho. Na segunda análise todo o conteúdo foi avaliado com o intuito de identificar a relação com as questões levantadas e com os critérios que necessitavam de uma leitura mais detalhada. Os trabalhos que não foram selecionados não afetam o resultado final, pois não atendem aos critérios mínimos. Conforme feito no planejamento, a execução da pesquisa foi conduzida com o auxílio da orientadora, com o intuito de minimizar as inconsistências. Outros fatores relacionados aos critérios de inclusão e exclusão podem acabar removendo um trabalho importante da pesquisa, como é o caso do período selecionado e do idioma, definidos como critérios obrigatórios para seleção.

Validade externa: A revisão sistemática não possui foco em nenhum tipo de estudo, ou seja, podem ser utilizados estudos primários, secundários ou até terciários, ela está delimitada conforme seus critérios de inclusão e exclusão conforme definido nas seções 3.1.3.2 e 3.1.3.1, este fator traz a ameaça de que, no momento de execução do trabalho, as informações encontradas não representem, em sua completude, os padrões utilizados no mercado. Esta ameaça é minimizada através do período utilizado, considerando dez anos de publicações, e por priorizar a relevância nas bases que apresentaram um número alto de resultados.

Validade de conclusão: No decorrer do processo da revisão sistemática, o planejamento e a execução eram constantemente revisados com o auxílio da orientadora, a fim de resolver quaisquer divergência e/ou discordância com relação ao trabalho desenvolvido.

Figura 7 – Seleção de estudos



Fonte: Elaboração do autor, 2018

3.1.4 Resumo e síntese de estudos relevantes

Após a execução das buscas e coleta dos resultados, foram executados dois processos, sintetizados na **Figura 7**.

1. Análise dos títulos e resumos:

Ao analisar os títulos e resumos, foram removidos todos artigos que não tinham relação com o tema proposto e que conseqüentemente não seriam relevantes para responder as questões abordadas no presente trabalho. Nesta etapa, dos 453 artigos analisados, foram selecionados 222 artigos únicos, cerca de 50% do montante geral. Esta etapa utiliza condições mais superficiais, como a análise do assunto abordado pelo trabalho e seu objetivo. Este processo é rápido e diminui consideravelmente o número de trabalhos que precisam de uma análise mais detalhada;

2. Análise completa:

Nesta etapa são avaliados todos os trabalhos restantes de uma maneira mais detalhada, considerando todo o conteúdo dos estudos. Todos os critérios de seleção são considerados e definem se um artigo será selecionado ou não. Dos 222 artigos que passaram pela primeira etapa, foram selecionados 30 artigos únicos, cerca de 13% do montante geral.

A execução e seleção dos artigos gerou insumos para responder as questões levantadas no planejamento da revisão sistemática. Estas questões estão descritas nas seções 3.1.4.1 e 3.1.4.2 deste capítulo.

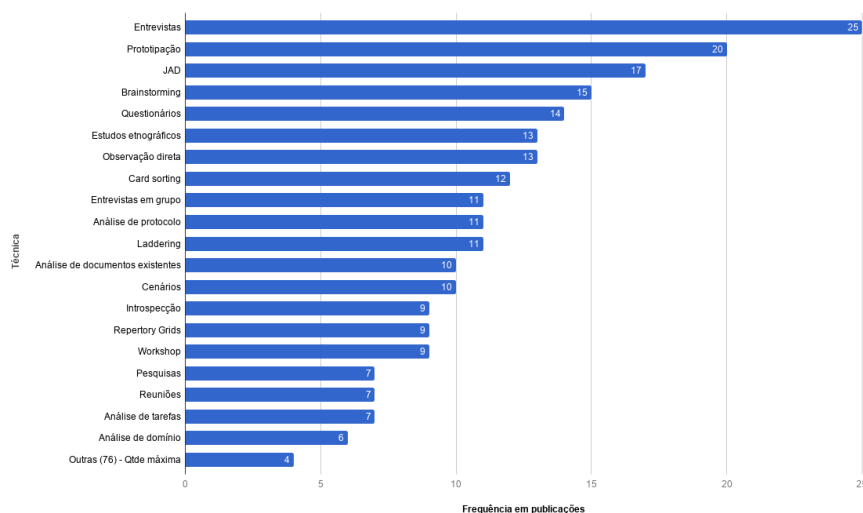
3.1.4.1 Q1 - Quais técnicas são utilizadas para elicitación de requisitos?

Esta questão tem o objetivo de identificar as técnicas mais relevantes através dos trabalhos selecionados na revisão sistemática e obter informações inerentes às mesmas. Não existe a pretensão de atingir todas as técnicas existentes no meio acadêmico ou na indústria.

Conforme apresentado na **Tabela 3**, foram levantadas 96 técnicas. O número coletado demonstra que existem muitas técnicas que são pouco exploradas, a maioria (59%) é citada apenas 1 vez. Cerca de 21% das técnicas são encontradas em mais de 5 publicações. O estudo comprova também que a *Entrevista* é a técnica mais comum, encontrada em cerca de 83% das publicações selecionadas. Algumas técnicas podem possuir nomes diferentes, mas fazem referência ao mesmo método. Os casos identificados como similares através dos textos dos próprios autores foram unificados, caso contrário as técnicas foram mantidas com seus nomes de origem.

Na **Figura 8** são apresentadas as técnicas mais encontradas (> 5 citações) através de um gráfico de barras. A **Figura 9** apresenta, através de uma nuvem de palavras, todas as técnicas encontradas, sendo que seus respectivos tamanhos na figura representam sua frequência perante as demais.

Figura 8 – Gráfico de técnicas por frequência nas publicações



Fonte: Elaboração do autor, 2018

Tabela 3 – Técnicas coletadas por publicação

#	Técnicas de elicitação de requisitos	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	
T01	Entrevistas individuais	X	X	X	X	X	X	X	X	-	X	X	X	-	X	X	X	X	X	X	X	X	X	X	X	X	-	-	X	X	X	-
T02	Entrevistas em grupo	X	-	X	-	X	X	-	X	-	X	-	-	X	-	-	X	X	-	-	-	X	-	-	-	-	-	-	-	-	-	X
T03	Prototipação	X	X	X	X	X	X	X	-	-	X	X	-	-	X	X	X	-	X	X	X	X	X	-	-	X	-	X	X	-	X	X
T04	JAD (Joint Application Design)	-	X	-	X	X	X	X	-	-	X	X	X	X	-	X	X	X	-	X	-	X	-	-	-	X	X	-	-	X	X	-
T05	FAST (Facilitated Application Specification Techniques)	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
T06	Introspecção	-	X	-	-	X	-	X	-	-	-	X	-	-	-	X	-	-	X	X	-	-	-	-	X	-	-	-	X	-	-	
T07	Questionários	-	X	-	X	X	-	X	-	-	X	X	-	-	-	X	-	-	X	X	X	X	-	-	X	X	-	-	X	X	-	-
T08	Análise de protocolo	-	X	X	-	X	-	X	-	-	-	X	-	-	-	-	X	-	X	X	-	X	X	-	-	-	-	-	-	X	-	-
T09	Análise de documentos existentes	-	X	-	-	X	X	X	-	-	X	-	-	-	-	-	X	-	-	-	X	-	-	-	X	-	-	-	-	-	X	X
T10	Estudos etnográficos	-	X	X	X	X	X	X	X	-	-	X	-	-	X	X	X	-	X	-	-	-	-	-	-	-	-	-	-	X	-	-
T11	Prototipação e experimentação	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
T12	Pesquisas	-	-	-	X	X	-	X	X	-	-	X	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	X	-
T13	Card sorting	-	-	-	X	X	X	X	-	-	-	X	-	-	-	-	X	-	X	X	X	X	X	-	X	X	-	-	X	-	-	-
T14	Laddering	-	-	-	X	X	X	X	-	-	-	X	-	-	-	-	X	-	X	X	X	-	X	X	-	-	-	-	-	X	-	-
T15	Repertory Grids	-	-	-	X	X	X	X	-	-	-	X	-	-	-	-	-	-	X	X	X	-	X	-	-	-	-	-	-	X	-	-
T16	Brainstorming	-	-	-	X	X	X	X	-	X	X	-	X	X	-	-	X	X	X	-	-	X	-	-	-	X	-	-	-	X	-	X
T17	Observação direta	-	-	-	X	X	X	-	X	-	X	X	-	-	-	-	X	-	X	X	X	X	X	-	-	-	-	-	-	X	-	X
T18	Observação indireta	-	-	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T19	Workshops	-	-	-	-	X	X	-	-	-	-	X	-	-	-	-	-	X	X	-	-	-	-	-	X	X	-	-	X	X	-	-
T20	Cenários	-	-	-	-	X	X	-	-	X	-	X	-	-	X	-	X	-	-	X	-	X	-	-	-	X	-	-	-	X	-	-
T21	CRC (Class Responsibility Collaboration)	-	-	-	-	X	-	-	-	-	-	X	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
T22	Reutilização de requisitos	-	-	-	-	-	X	-	-	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T23	RAD (Rapid Application Development)	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
T24	Reuniões	-	-	-	-	-	-	X	-	-	-	X	-	X	X	-	-	-	X	-	-	-	-	-	X	-	-	-	X	-	-	-
T25	Análise de discurso	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
T26	SSM (Soft Systems Methodology)	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
T27	Técnica de escala de proximidade	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T28	Análise de tarefas	-	-	-	-	-	-	X	-	-	-	X	-	-	-	-	-	-	X	X	-	X	-	-	X	-	-	-	X	-	-	-
T29	Estudo de caso	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T30	Relatório experimental	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T31	EPMcreate (Elementary Pragmatic Model)	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
T32	POEPMcreate (Power-OnlyEPMcreate)	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T33	Análise de interface	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T34	Análise de domínio	-	-	-	-	-	-	-	-	-	-	X	-	-	X	-	-	-	X	-	-	-	-	-	X	X	-	-	X	-	-	-
T35	Casos de uso	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	X	-	-	-	X	X	-	-	-	-	-
T36	WinWin	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
T37	EasyWinWin	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T38	CoREA (Collaborative Requirements Elicitation and Analysis)	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T39	Athena	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T40	Modelagem de metas	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	X	-	-	-	X	-	-	-
T41	Viewpoints	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	X	-	-	-	X	-	-	-
T42	Storyboarding	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T43	Investigação contextual	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T44	Investigação apreciativa	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T45	Diagramação	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
T46	Diagrama de fluxo de dados	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
T47	Mapa de conceito	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
T48	Processo orientado a informação	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
T49	Entrevistas baseadas em testes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	X
T50	20 questões	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
T51	Descrição ideal	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-

Tabela 4 – Características relacionadas às técnicas de elicitação de requisitos

#	Característica	Descrição	Opções de valores	Publicação
C01	Tipo de técnica	Identifica qual a forma de extração de requisitos	Direta ou Indireta	(ABBASI et al., 2015)
C02	Tipo de dado	Identifica o tipo do dado extraído	Qualitativo ou Quantitativo	(ABBASI et al., 2015)
C03	Comunicação	Identifica a direção da comunicação entre os envolvidos	Unidirecional ou Bidirecional	(ABBASI et al., 2015)
C04	Compreensão do domínio	Identifica a necessidade de entendimento do domínio da aplicação	Sim ou Não	(ABBASI et al., 2015)
C05	Fonte principal	Fonte principal responsável por fornecer insumos ou conduzir a técnica	Analista com conhecimento no domínio, Facilitador externo, Observador, Documentação, Especialista ou Analistas e Stakeholders	(MRAYAT; NORWAWI; BASIR, 2013)
C06	Treinamento na técnica de elicitação	Treinamento prévio e prática do elicitor com a técnica de elicitação	Alto, Baixo ou Nenhum	(CARRIZO; DIESTE; JURISTO, 2014)
C07	Experiência do elicitor	Número de projetos que o elicitor realizou atividades de elicitação	Alto, Médio ou Baixo	(CARRIZO; DIESTE; JURISTO, 2014)
C08	Experiência com técnicas de elicitação	Número de projetos que o elicitor realizou atividades com determinada técnica	Alto, Baixo ou Nenhum	(CARRIZO; DIESTE; JURISTO, 2014)
C09	Familiaridade com o domínio	Número de projetos no mesmo domínio ou no domínio de conhecimento do elicitor	Alto, Baixo ou Nenhum	(CARRIZO; DIESTE; JURISTO, 2014)
C10	Pessoas por sessão	Número de indivíduos que podem participar da sessão de elicitação (Individual/Grupo/Em massa)	Individual, Grupo ou Em massa	(CARRIZO; DIESTE; JURISTO, 2014)
C11	Consenso entre os stakeholders	Acordo inicial entre os informantes (Alto/Baixo)	Alto ou Baixo	(CARRIZO; DIESTE; JURISTO, 2014)
C12	Interesse do stakeholder	Interesse do informante em participar das sessões de elicitação	Alto, Baixo ou Nenhum	(CARRIZO; DIESTE; JURISTO, 2014)
C13	Especialidade	Experiência do informante no problema ou no domínio	Especialista, Bem informado ou Iniciante	(CARRIZO; DIESTE; JURISTO, 2014)
C14	Articulação	Habilidade do informante em passar seu conhecimento	Alto, Médio ou Baixo	(CARRIZO; DIESTE; JURISTO, 2014)
C15	Disponibilidade de tempo	Tempo disponível do informante	Alto ou Baixo	(CARRIZO; DIESTE; JURISTO, 2014)
C16	Local/Acessibilidade	Localização do informante em relação ao elicitor	Longe ou Perto	(CARRIZO; DIESTE; JURISTO, 2014)
C17	Tipo de informação a elicitar	Tipo de informação que a técnica pode levantar	Estratégica, Tática ou Básica	(CARRIZO; DIESTE; JURISTO, 2014)
C18	Nível de informação disponível	Tipo de categorização de informação disponível antes da execução da técnica	Superior, Inferior ou Nenhum	(CARRIZO; DIESTE; JURISTO, 2014)
C19	Definição do problema	Clareza dos objetivos e escopo do projeto	Alto ou Baixo	(CARRIZO; DIESTE; JURISTO, 2014)
C20	Restrição de tempo do projeto	Tempo disponível para aplicação da técnica no projeto	Alto, Médio ou Baixo	(CARRIZO; DIESTE; JURISTO, 2014)
C21	Tempo de processo	Etapa de pré-sessão do processo de elicitação	Início, Meio ou Fim	(CARRIZO; DIESTE; JURISTO, 2014)

Fonte: Elaboração do autor, 2018

3.1.5 Determinando a aplicabilidade dos resultados

O número de técnicas coletadas e suas respectivas características representam a validade do planejamento da revisão sistemática. Estas informações fornecem insumos para que as técnicas possam ser devidamente classificadas perante suas características. Estes dados também mostram que a área de Elicitação de Requisitos está em constante de evolução, surgindo novas técnicas e derivações das técnicas existentes. Os métodos consolidados também são citados pela academia em vários dos artigos encontrados.

3.1.6 Revisar e avaliar a literatura

Os artigos selecionados representam, na maioria dos casos, estudos secundários. Estes casos foram selecionados através de uma revisão sistemática similar ao presente trabalho. Não foi encontrado na literatura, um artigo que tenha o mesmo objetivo deste trabalho, dificultando a comparação e validação do processo.

4 CLASSIFICAÇÃO

Este capítulo tem o objetivo de descrever o processo de criação das facetas conforme citado na **Sessão 2.2.3** do **Capítulo 2**. Ao final deve-se ter como resultado uma versão final das facetas definidas e a aplicação da classificação facetada em parte das técnicas selecionadas.

4.1 COLETA DE DOMÍNIOS

O domínio será representado por todas as 96 técnicas coletadas através da revisão sistemática (**Capítulo 3**) e apresentadas na **Tabela 3**. As técnicas foram agrupadas conforme sua categoria e são apresentadas nas **Tabelas 5, 6, 7, 8 e 9**.

Tabela 5 – Técnicas tradicionais (ABBASI et al., 2015) (YOUSEF; ALMARABEH, 2015) (SHARMA; PANDEY, 2013) (REHMAN; KHAN; RIAZ, 2013) (SHARMA; PANDEY, 2014)

Técnicas presentes neste grupo			
Entrevistas	Questionários	Análise de documentos existentes	Reuniões
Entrevistas em grupo	Pesquisas	Introspecção	Análise de domínio

Fonte: Elaboração do autor, 2018

Tabela 6 – Técnicas cognitivas (ABBASI et al., 2015) (YOUSEF; ALMARABEH, 2015) (SHARMA; PANDEY, 2013)

Técnicas presentes neste grupo			
Card sorting	Repertory Grids	Análise de protocolo	Análise de tarefas
Laddering	CRC	Técnica de escala de proximidade	

Fonte: Elaboração do autor, 2018

Tabela 7 – Técnicas Inovadoras (SHARMA; PANDEY, 2013)

Técnicas presentes neste grupo
Mockups em papel

Fonte: Elaboração do autor, 2018

Tabela 8 – Técnicas de grupo/colaborativas/criativas (YOUSEF; ALMARABEH, 2015) (YOUSEF; ALMARABEH, 2015) (SHARMA; PANDEY, 2013) (SAHA et al., 2012) (REHMAN; KHAN; RIAZ, 2013) (DUARTE et al., 2012) (NIJEM, 2013) (GHANBARI; SIMILÄ; MARKKULA, 2015)

Técnicas presentes neste grupo			
JAD	EPMcreate	Estruturação	Casos de uso
Prototipação	POEPMcreate	Conhecimento de especialista	WinWin
Prototipação e experimentação	Mapa mental	Entrada aleatória	EasyWinWin
Brainstorming	Brainsketching	Seis chapéus de pensamento	CoREA
Workshops	Brainwriting	Técnica da pausa criativa	Athena
Cenários	Checklists	5W1H	RESCUE
RAD	Perspectivas diferentes	Cenários futuros	Essence

Fonte: Elaboração do autor, 2018

Tabela 9 – Técnicas não categorizadas

Técnicas presentes neste grupo			
FAST	Mapa de conceito	Hierarquia top-down multi-atributo	Ontologias
Reutilização de requisitos	Processo orientado a informação	Decomposição de metas top-down	Aprendizado
Estudo de caso	Entrevistas baseadas em testes	Grupo nominal	Análise de Stakeholder
Relatório experimental	20 questões	Delphi	Demonstração de tarefa
Análise de interface	Descrição ideal	QFD	Companias similares
Modelagem de metas	Imagem negativa	Designer como aprendizado	Categorização
Viewpoints	Imagem positiva	Histórias de usuários	Hierarquia analítica de processos
Storyboarding	Escala multidimensional	IBIS	Métodos de \$100
Investigação contextual	Classificação livre	Misuse cases	Jogo de planejamento
Investigação apreciativa	Escala hierárquica	Pseudo software	Inspeção
Diagramação	Ranking	Expressão de requisitos controlada	Revisão
Diagrama de fluxo de dados	Listagem de atributos bottom-up	ARM	

Fonte: Elaboração do autor, 2018

A nomenclatura das técnicas e suas respectivas categorias foram extraídas diretamente das publicações selecionadas, conforme exposto na **Tabela 3**.

4.2 LISTAGEM DE ENTIDADES

As entidades estão devidamente listadas na **Sessão 4.1**, onde cada técnica de elicitação de requisitos, coletada através da revisão sistemática realizada e documentada no **Capítulo 3**, representa uma entidade nesta classificação.

4.3 CRIAÇÃO DE FACETAS

Para Denton (2003), na criação das facetas deve-se examinar os termos resultantes e analisar quais categorias gerais de alto nível aparecem em todas as entidades. Neste caso,

todas as características e seus respectivos valores foram extraídos das publicações selecionadas na revisão sistemática e expostas na **Tabela 4**. As categorias macro, utilizadas na **Sessão 4.1** também formam uma das facetas.

Desta forma, as seguintes características devem derivar facetas para classificar as técnicas previstas: Categoria, Tipo de técnica, Tipo de dado, Comunicação, Fonte principal, Treinamento na técnica de elicitação, Experiência do elicitor, Experiência com técnicas de elicitação, Familiaridade com o domínio, Pessoas por sessão, Consenso entre os stakeholders, Interesse do stakeholder, Especialidade, Articulação, Disponibilidade de tempo, Local/Acessibilidade, Tipo de informação a elicitar, Nível de informação disponível, Definição do problema, Restrição de tempo do projeto e Tempo de processo.

4.4 ARRANJO DE FACETAS

As facetas serão organizadas alfabeticamente. Neste caso, e conforme foi exemplificado por (DENTON, 2003), é a maneira mais simples e apropriada pois a ordem não vai auxiliar os usuários na identificação (**Tabela 10**).

Tabela 10 – Arranjo de facetas

Faceta	Valores
Categoria	<i>Cognitiva, Contextual, Grupo, Inovadora, Tradicional</i>
Fonte principal	<i>Analista com conhecimento no domínio, Analistas e Stakeholders, Documentação, Especialista, Facilitador externo, Observador</i>
Treinamento na técnica de elicitação	<i>Alto, Baixo, Nenhum</i>
Experiência do elicitor	<i>Alto, Baixo, Médio</i>
Experiência com técnicas de elicitação	<i>Alto, Baixo, Nenhum</i>
Familiaridade com o domínio	<i>Alto, Baixo, Nenhum</i>
Tipo de informação a elicitar	<i>Básica, Estratégica, Tática</i>
Nível de informação disponível	<i>Inferior, Nenhum, Superior</i>
Definição do problema	<i>Alto, Baixo</i>
Tipo de dado	<i>Qualitativo, Quantitativo</i>
Comunicação	<i>Bidirecional, Unidirecional</i>
Pessoas por sessão	<i>Em massa, Grupo, Individual</i>
Consenso entre os stakeholders	<i>Alto, Baixo</i>
Interesse do stakeholder	<i>Alto, Baixo, Nenhum</i>
Especialidade	<i>Especialista, Bem informado, Iniciante</i>
Articulação	<i>Alto, Baixo, Médio</i>
Disponibilidade de tempo	<i>Alto, Baixo</i>
Local/Acessibilidade	<i>Longe, Perto</i>
Tipo de técnica	<i>Direta, Indireta</i>
Restrição de tempo do projeto	<i>Alto, Baixo, Médio</i>
Tempo de processo	<i>Fim, Início, Meio</i>

Fonte: Elaboração do autor, 2018

4.5 ORDEM DE CITAÇÃO

Para facilitar o entendimento das facetas criadas, foi desenvolvido um agrupamento para cada conjunto de itens com base em seu contexto de aplicação. A ordem das facetas

e a maneira de apresentação adotada é apresentada na **Tabela 11**, conforme o índice de cada característica. Assim como o agrupamento de facetas, essa ficha tem o objetivo de sintetizar toda a classificação de uma técnica. Apesar da ordem não afetar o resultado final da classificação, alguns fatores como a capacidade de agrupamento em relação as técnicas ou o assunto relacionado à característica foram levados em consideração. Os dados apresentados com o conteúdo *valor* devem ser substituídos pelas informações citadas na **Seção 4.4**.

Tabela 11 – Modelo de organização das facetas

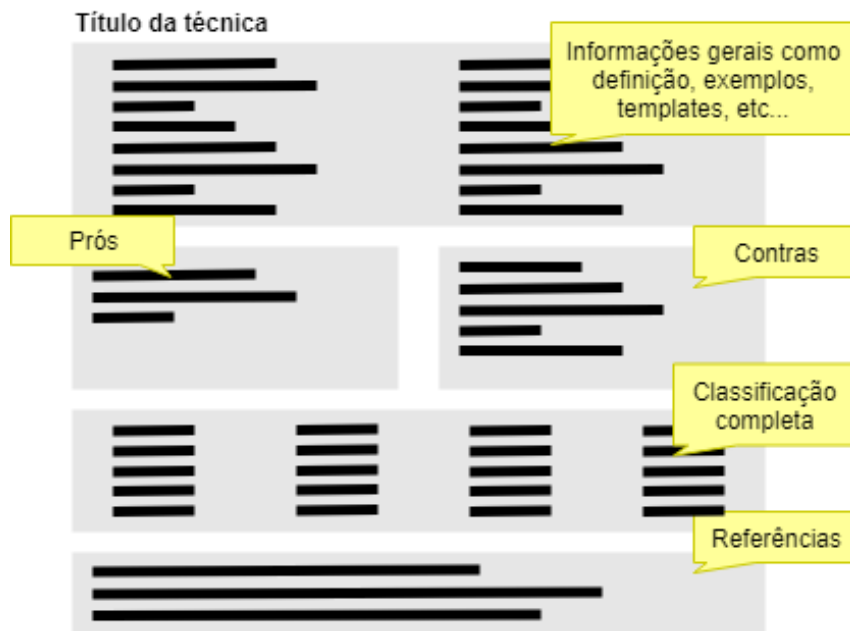
Técnica		<i>valor</i>	
1. Categoria		<i>valor</i>	
Elicitor		2. Fonte principal	<i>valor</i>
3. Treinamento na técnica de elicitação	<i>valor</i>	Domínio do problema	
4. Experiência do elicitor	<i>valor</i>	7. Tipo de informação a elicitar	<i>valor</i>
5. Experiência com técnicas de elicitação	<i>valor</i>	8. Nível de informação disponível	<i>valor</i>
6. Familiaridade com o domínio	<i>valor</i>	9. Definição do problema	<i>valor</i>
		10. Tipo de dado	<i>valor</i>
		11. Comunicação	<i>valor</i>
Stakeholder		Características gerais da técnica	
12. Pessoas por sessão	<i>valor</i>	19. Tipo de técnica	<i>valor</i>
13. Consenso entre os stakeholders	<i>valor</i>	20. Restrição de tempo do projeto	<i>valor</i>
14. Interesse do stakeholder	<i>valor</i>	21. Tempo de processo	<i>valor</i>
15. Especialidade	<i>valor</i>		
16. Articulação	<i>valor</i>		
17. Disponibilidade de tempo	<i>valor</i>		
18. Local/Acessibilidade	<i>valor</i>		

Fonte: Elaboração do autor, 2018

4.6 CLASSIFICAÇÃO

Nesta sessão são apresentadas as técnicas e suas devidas parametrizações quanto às facetas criadas. As **Seções 4.6.1 e 4.6.2** trazem exemplos práticos da classificação aplicada ao domínio definido. Ao todo, para o presente trabalho, foram classificadas 12 técnicas das 96 coletadas na revisão sistemática. O critério utilizado para priorizar a classificação foi o percentual de facetas encontradas para cada técnica. Este critério foi utilizado para viabilizar a execução do trabalho. Neste capítulo são apresentadas duas das classificações feitas e o restante das técnicas classificadas é acessível através do guia desenvolvido como parte da proposta de trabalho. O link de acesso ao Guia é apresentado na **Seção 5.4.2**. A estrutura utilizada para apresentação do conteúdo segue um padrão para todas as classificações, conforme exemplificado na **Figura 10**.

Figura 10 – Organização do conteúdo apresentado no guia



Fonte: Elaboração do autor, 2018

4.6.1 Entrevistas

É conhecida por ser a mais comum das técnicas de eliciação de requisitos (ZOWGHI; COULIN, 2005). As entrevistas geralmente envolvem discussões individuais de um representante da equipe com o stakeholder do novo sistema (HANSEN; BERENTE; AVITAL, 2014). Devido à capacidade desta técnica para obter conhecimento profundo, é considerada como técnica importante para obter e validar os requisitos de software (YOUSEF; ALMARABEH, 2015). É um método verbal, considerado fácil e efetivo para compartilhar ideias e expressar necessidades entre os analistas e os stakeholders. Através de uma conversa, são feitas perguntas e são documentados os fatores que podem estar associados aos requisitos (YOUSUF; ASGER, 2015). As principais características dos Analistas são:

- Estar abertos a novas ideias, evitam ideias preconcebidas sobre os requisitos e estão dispostos a ouvir os stakeholders, mesmo que o stakeholder apresente requisitos-surpresa, eles estão dispostos a mudar de ideia sobre o sistema.
- Eles estimulam o entrevistado a participar de discussões com uma questão-trampolim, uma proposta de requisitos ou trabalhando em conjunto em um protótipo do sistema. É improvável que dizer às pessoas "diga-me o que quiser" resulte em informações úteis. É muito mais fácil falar em um contexto definido do que em termos gerais.

Geralmente as entrevistas são separadas em:

- **Estruturadas:** Entrevistas estruturadas são baseadas em um conjunto fixo de perguntas, juntamente com informações detalhadas. Este tipo permite que os analistas examinem o nível de compreensão que um stakeholder tem sobre determinado tópico. Templates também são parte deste processo que fornece um caminho mais fácil para a elicitación de requisitos (SHARMA; PANDEY, 2013).
- **Não estruturadas:** Entrevistas não estruturadas são conversacionais por natureza, onde não é necessário preparar perguntas, e as informações dos stakeholders estão em discussões abertas (SHARMA; PANDEY, 2013). São mais úteis quando você quer se concentrar na compreensão de um determinado problema dentro do ambiente que o usuário está inserido (ZHANG; WILDEMUTH, 2016).
- **Semi-estruturada:** É uma combinação de perguntas pré-definidas e não planejadas. Em outras palavras, é uma mistura dos tipos estruturado e não estruturado (YOUSUF; ASGER, 2015).

4.6.1.1 Prós

- Fácil extrair os detalhes fazendo perguntas de acompanhamento (MRAYAT; NORWAWI; BASIR, 2013).
- Identifica sentimentos e objetivos de diferentes indivíduos (MRAYAT; NORWAWI; BASIR, 2013) (YOUSUF; ASGER, 2015).
- É boa para tratar de tópicos complexos (YOUSUF; ASGER, 2015).
- Coleta informações importantes e detalhadas (YOUSUF; ASGER, 2015).
- As ambiguidades são esclarecidas (YOUSUF; ASGER, 2015).
- Diminui a chance de coletar informações falsas dos stakeholders (YOUSUF; ASGER, 2015).
- A falta de respostas é baixa (YOUSUF; ASGER, 2015).
- Fornece uma visão geral de todo o sistema (YOUSUF; ASGER, 2015).
- Coleta de informações que podem formar uma pesquisa ou outra atividade (ARIF; GAHYUR, 2009).

- Mais eficaz do que as outras técnicas na maioria dos casos, mesmo não sendo sempre a mais eficiente (DIESTE; JURISTO; SHULL, 2008).

4.6.1.2 Contrás

- Podem dar a falsa impressão de coleta de informação dos stakeholders dependendo da abordagem feita pelo analista ao fazer os questionamentos (HANSEN; BERENTE; AVITAL, 2014). Esta característica está relacionada com as limitações impostas pela linha de questionamento do entrevistador.
- Podem captar somente parte das práticas de trabalho que foram expostas pelos stakeholders. (DAVIS, 1982).
- As atividades descritas pelos stakeholders podem diferir do que realmente acontece (GOGUEN; LINDE, 1993).
- O sucesso depende muito das habilidades do entrevistador (MRAYAT; NORWAWI; BASIR, 2013).
- Não existe garantia na obtenção de informações significativas (MRAYAT; NORWAWI; BASIR, 2013).
- A linguagem pode ser uma barreira entre o entrevistador e os stakeholders (ISABIRYE; FLOWERDAY, 2008).
- Poucas pessoas envolvidas (YOUSUF; ASGER, 2015).
- É difícil definir o tempo adequado para entrevistas com todas pessoas (YOUSUF; ASGER, 2015).
- As informações não podem ser coletadas de muitas pessoas geralmente, pois aumenta o custo do processo (YOUSUF; ASGER, 2015).
- São trabalhosas e demoradas (YOUSUF; ASGER, 2015).
- Algumas vezes é necessário esclarecer alguma informação e os participantes podem não lembrar das informações coletadas anteriormente (YOUSUF; ASGER, 2015).

A **Tabela 12** apresenta a classificação facetada associada à técnica de elicitação intitulada de Entrevista.

Tabela 12 – Classificação - Entrevista

Técnica		Entrevista	
1. Categoria	Tradicional	2. Fonte principal	Analista com conhecimento no domínio
Elicitor		Domínio do problema	
3. Treinamento na técnica de elicitação	Baixo	7. Tipo de informação a elicitar	Básica, Tática
4. Experiência do elicitor	Médio	8. Nível de informação disponível	Inferior, Superior
5. Experiência com técnicas de elicitação	Baixo	9. Definição do problema	Alto
6. Familiaridade com o domínio	Baixo	10. Tipo de dado	Qualitativo, Quantitativo
		11. Comunicação	Bidirecional
Stakeholder		Características gerais da técnica	
12. Pessoas por sessão	Individual	19. Tipo de técnica	Direta
13. Consenso entre os stakeholders	Alto	20. Restrição de tempo do projeto	Baixo
14. Interesse do stakeholder	Alto	21. Tempo de processo	Fim, Meio
15. Especialidade	Especialista, Bem informado		
16. Articulação	Médio		
17. Disponibilidade de tempo	Alto		
18. Local/Acessibilidade	Perto		

Fonte: Elaboração do autor, 2018

4.6.2 Prototipação

Prototipação pode ser considerada como a confecção de uma versão inicial do produto, preparada para coletar o feedback das partes interessadas e levantar alterações que devem ser incorporadas na próxima edição (ABBASI et al., 2015). O uso de protótipos é muito útil quando os stakeholders não tem muita familiaridade com as opções tecnológicas disponíveis para atender sua necessidade (ZOWGHI; COULIN, 2005). Uma visão antecipada do software torna fácil a determinação de ambiguidades ou conflitos nos requisitos. É importante ressaltar que apesar de causar uma sobrecarga, o protótipo diminui as chances de falha da concepção, uma garantia de 80% de seleção dos requisitos apropriados (KHAN et al., 2011). Desta forma, esta técnica visa descobrir e visualizar os requisitos da interface antes de projetar ou desenvolver o aplicativo (YOUSEF; ALMARABEH, 2015). É útil para o desenvolvimento de novos sistemas, principalmente quando a plataforma tem mais interações de usuário do que funcionalidades internas ou quando os stakeholders não estão familiarizados com as soluções disponíveis (YOUSUF; ASGER, 2015) (MRAYAT; NORWAWI; BASIR, 2013). Vem sendo bastante utilizada para elicitação onde há muita incerteza nos requisitos ou quando o stakeholder precisa do feedback (DAVIS, 1992).

Podem fornecer dois tipos de requisitos (LAUESEN, 2017):

- **Relacionados ao produto:** Define uma funcionalidade requerida, é realista e útil.
- **Relacionados ao design:** Define como a interface do sistema será apresentada.

Na elicitação de requisitos os tipos de protótipos podem ser divididos em (VIANNA,

2012):

- **Protótipo em papel:** são representações de interfaces gráficas com diferentes níveis de fidelidade, desde um wireframe desenhado à mão em pequenos pedaços de papel até uma embalagem de sabonete.
- **Protótipo de experiência:** é a simulação da experiência do serviço que prevê alguns de seus desempenhos através do uso de interações específicas envolvidas.

4.6.2.1 Prós

- Envolvimento do usuário durante o processo de desenvolvimento (YOUSUF; ASGER, 2015).
- Permite o feedback antecipado do usuário para o refinamento de requisitos (YOUSUF; ASGER, 2015).
- Economiza tempo e custo de desenvolvimento (YOUSUF; ASGER, 2015).
- Usuário e analistas entendem melhor o sistema (YOUSUF; ASGER, 2015).
- Com a prototipagem, não há requisitos errados, mas somente aqueles que esperam para ser descobertos (TOMAYKO, 2002).
- Podem demonstrar o progresso desde o estágio inicial de desenvolvimento (KHAN¹; DULLOO; VERMA, 2014).
- Acrescenta qualidade e comunicação entre os analistas e usuários (KHAN¹; DULLOO; VERMA, 2014).
- Resulta em um alto nível de satisfação de usuário (GUNDA, 2008).

4.6.2.2 Contras

- Pode elevar a expectativa dos usuários, influenciando negativamente na recepção do produto final, criando uma resistência às mudanças efetuadas até a versão final (HANSEN; BERENTE; AVITAL, 2014) (YOUSUF; ASGER, 2015).
- Quando um protótipo não é descartado, ou seja, é usado para dar segmento ao desenvolvimento do produto, ele pode acabar gerando um software que dificulta a manutenção (HANSEN; BERENTE; AVITAL, 2014).

- A estimativa de esforço e custo pode ser alta se for feita muito cedo (YOUSUF; ASGER, 2015).
- Pode ser demorado para sistemas complexos (YOUSUF; ASGER, 2015).
- O desenvolvimento de múltiplos protótipos de alta fidelidade pode se tornar caro (INAYAT et al., 2015).
- Às vezes leva a documentação incompleta (KHAN¹; DULLOO; VERMA, 2014).

A **Tabela 13** apresenta a classificação facetada associada à técnica de elicitação intitulada de Prototipação.

Tabela 13 – Classificação - Prototipação

Técnica		Prototipação	
1. Categoria	Grupo	2. Fonte principal	Analistas e Stakeholders
Elicitor		Domínio do problema	
3. Treinamento na técnica de elicitação	Alto	7. Tipo de informação a elicitar	Básica, Tática
4. Experiência do elicitor	Médio	8. Nível de informação disponível	Inferior, Superior
5. Experiência com técnicas de elicitação	Baixo	9. Definição do problema	Baixo
6. Familiaridade com o domínio	Alto	10. Tipo de dado	Qualitativo
		11. Comunicação	Bidirecional
Stakeholder		Características gerais da técnica	
12. Pessoas por sessão	Individual	19. Tipo de técnica	Direta
13. Consenso entre os stakeholders	Baixo	20. Restrição de tempo do projeto	Baixo
14. Interesse do stakeholder	Nenhum	21. Tempo de processo	Meio
15. Especialidade	Iniciante		
16. Articulação	Baixo		
17. Disponibilidade de tempo	Alto		
18. Local/Acessibilidade	Perto		

Fonte: Elaboração do autor, 2018

4.7 REVISÃO, TESTE E MANUTENÇÃO

A etapa de revisão e teste deve ser executada após o desenvolvimento do guia e da classificação das técnicas de elicitação de requisitos. O acesso à ferramenta com as devidas informações cadastradas possibilitam avaliar se a definição das facetas foram adequadas e atendem aos interesses do público alvo e do domínio em questão.

4.8 CONSIDERAÇÕES FINAIS

Apesar de ser um processo tecnicamente simples, a criação de uma classificação facetada se mostra complexa em alguns pontos. Existe uma dependência muito grande dos insumos utilizados na categorização do domínio, além de ser necessário um certo conhecimento na área e nos objetivos da classificação para saber como tratar a informação

coletada. Outro fator muito importante está associado ao processo de teste e manutenção da estrutura da classificação facetada, que tende a refinar todo o trabalho desenvolvido impactando nos resultados alcançados através dos filtros.

5 DESENVOLVIMENTO

Esta seção possui o objetivo de especificar todas as decisões tomadas no desenvolvimento do guia. São apresentados os requisitos levantados, os casos de uso e seus respectivos detalhamentos, algumas telas da aplicação e outros artefatos que possuem o objetivo de facilitar a compreensão geral proposta.

5.1 REQUISITOS

Os requisitos foram elicitados segundo os aspectos funcionais (RF), não funcionais (RNF) e de negócio (RN), apresentados respectivamente nas **Seções 5.1.1, 5.1.2 e 5.1.3**. Para a elicitação de requisitos do presente trabalho, foram utilizadas as técnicas:

- Entrevistas: Para coletar os requisitos do projeto com a orientadora.
- Prototipação: Para validar as funcionalidades da aplicação e prever a organização as informações apresentadas.
- Análise de documentos: Para a coleta de informações utilizadas no Guia através da revisão sistemática apresentada no **Capítulo 3**.

5.1.1 Requisitos funcionais

Para os requisitos funcionais, foram priorizadas as ações do usuário final, das quais o objetivo principal é buscar e visualizar informações através do guia. Para tanto, seguem os requisitos levantados. Também foram levantados requisitos associados à gestão do conteúdo do guia.

- RF01: O sistema deve possibilitar que os usuários visualizem as classificações criadas.
- RF02: O sistema deve possibilitar que os usuários visualizem as técnicas de elicitação publicadas no guia.
- RF03: O sistema deve possibilitar que os usuários visualizem as facetas utilizadas na classificação acessada através do guia.
- RF04: O sistema deve possibilitar que os usuários efetuem buscas de técnicas com base na classificação disponibilizada.

- RF05: O sistema deve possibilitar que os usuários compartilhem a página do guia nas redes sociais.
- RF06: O sistema deve possibilitar que o administrador gerencie o cadastro das técnicas.
- RF07: O sistema deve possibilitar que o administrador gerencie o cadastro das facetas.
- RF08: O sistema deve possibilitar que o administrador gerencie a classificação das técnicas.
- RF09: O sistema deve possibilitar que o administrador gerencie os usuários que possuem acesso ao painel administrativo.
- RF10: O sistema deve possibilitar que o administrador efetue o login no painel administrativo.
- RF11: O sistema deve possibilitar que o administrador efetue o logout do painel administrativo.

5.1.2 Requisitos não funcionais

Além das questões associadas a apresentação do guia na web, é importante ressaltar que um fator determinante para a identificação dos requisitos não funcionais foi o conhecimento prévio em algumas das tecnologias utilizadas, como por exemplo a linguagem de programação utilizada na camada backend, este fator tem o objetivo de viabilizar o desenvolvimento do projeto em tempo hábil. Desta forma, os requisitos não funcionais listados levam em consideração os recursos necessários para a concepção do guia.

- RNF01: O guia deve ser disponibilizado como um website de acesso público.
- RNF02: O guia deve ser acessível dos navegadores Google Chrome (versão 67 ou superior) e Mozilla Firefox (versão 60 ou superior).
- RNF03: O conteúdo das páginas apresentadas deve aceitar texto puro, notação HTML e notação Markdown.
- RNF04: Para a camada do servidor, o guia deve ser desenvolvido em PHP utilizando a versão 7.1.

- RNF05: Para a camada do cliente, o guia deve ser desenvolvido utilizando HTML, CSS e Javascript.
- RNF06: Os dados devem ser persistidos em um banco de dados MySQL.
- RNF07: As redes sociais disponíveis para compartilhamento são: Facebook, Twitter e LinkedIn.

5.1.3 Requisitos de negócio

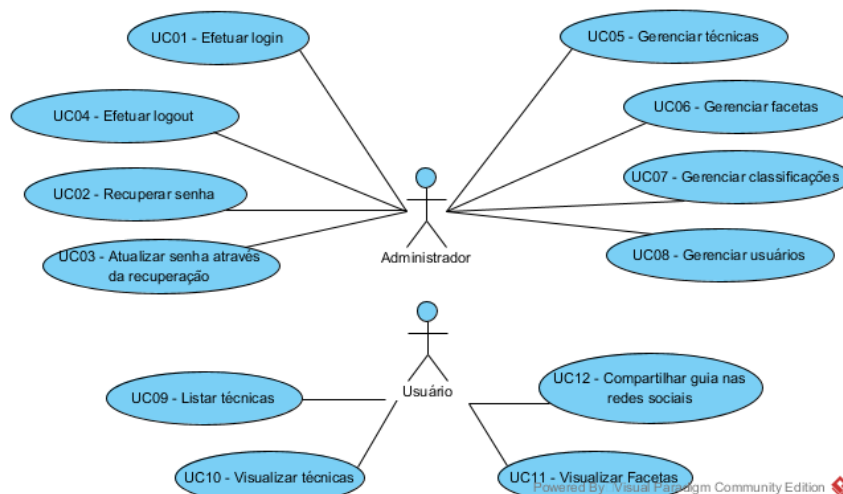
Com relação aos requisitos de negócio, as especificidades estão mais relacionadas aos filtros e sua execução.

- RN01: Deve ser possível controlar a junção entre os filtros selecionados, tornando a busca mais restritiva ou mais aberta (OU lógico ou E lógico).

5.1.4 Casos de uso

O caso de uso é uma técnica utilizada para definir o comportamento de um sistema, ou seja, ele descreve as interações entre o meio externo (usuários, aplicações, etc) e o meio interno. Sua representação através de um diagrama UML é muito importante para especificar, visualizar e documentar estes comportamentos (SENGUPTA; BHATTACHARYA, 2008). A **seção ??** apresenta um exemplo de detalhamento de um dos casos de uso previstos para o guia proposto. Todos os casos de uso apresentados na **Figura 11** estão detalhados no **Apêndice C**.

Figura 11 – Diagrama de casos de uso



Fonte: Elaboração do autor, 2018

5.2 UC09 - LISTAR TÉCNICAS

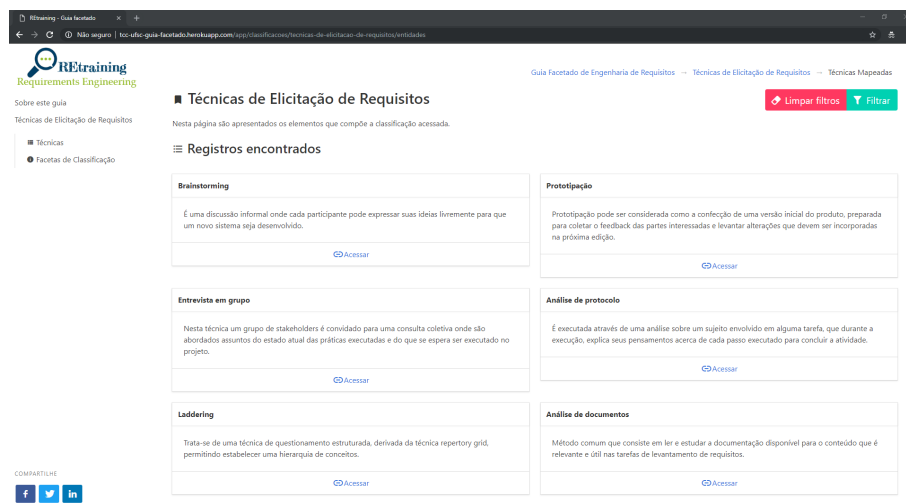
Lista as técnicas publicadas no guia.

Cenários

Visualiza técnicas publicadas (Figura 37) (*principal*)

1. O usuário acessa a opção *Técnicas* no menu principal.
2. O sistema apresenta as técnicas publicadas.

Figura 12 – Listar técnicas através do Guia



Fonte: Elaboração do autor, 2018

Apresenta mensagem caso não encontrar técnicas com base nos filtros (*alternativo*)

No passo 1, se o usuário executar a ação *Limpar filtros*, o sistema deve remover os filtros selecionados e apresentar as técnicas publicadas.

Visualiza técnicas publicadas com base nos filtros (Figura 38) (*alternativo*).

1. O usuário executa a opção *Filtrar* na listagem de técnicas.
2. O sistema apresenta os filtros disponíveis
3. O usuário seleciona os filtros.
4. O usuário executa a opção *Filtrar*.
5. O sistema busca as técnicas conforme os filtros selecionados.

6. O sistema apresenta as técnicas encontradas.

Figura 13 – Página de filtros disponíveis

Seleção de filtros conforme as seguintes facetas:

Geral

Categoria:

Fonte principal:

Elicitor

Treinamento na técnica de elicitação:

Experiência do elicitor:

Experiência com técnicas de elicitação:

Familiaridade com o domínio:

Domínio do problema

Tipo de dado:

Comunicação:

Tipo de informação a elicitar:

Nível de informação disponível:

Definição do problema:

Stakeholder

Pessoas por sessão:

Consenso entre os stakeholders:

Interesse do stakeholder:

Especialidade:

Articulação:

Disponibilidade de tempo:

Local/Acessibilidade:

Características gerais da técnica

Tipo de técnica:

Restrição de tempo do projeto:

Tempo de processo:

Fonte: Elaboração do autor, 2018

Apresenta mensagem caso não encontrar técnicas com base nos filtros (exceção)

No passo 4, dado que o usuário utilizou algum filtro, se o sistema não encontrar técnicas, deve apresentar a mensagem *Nenhum registro encontrado para o filtro informado*.

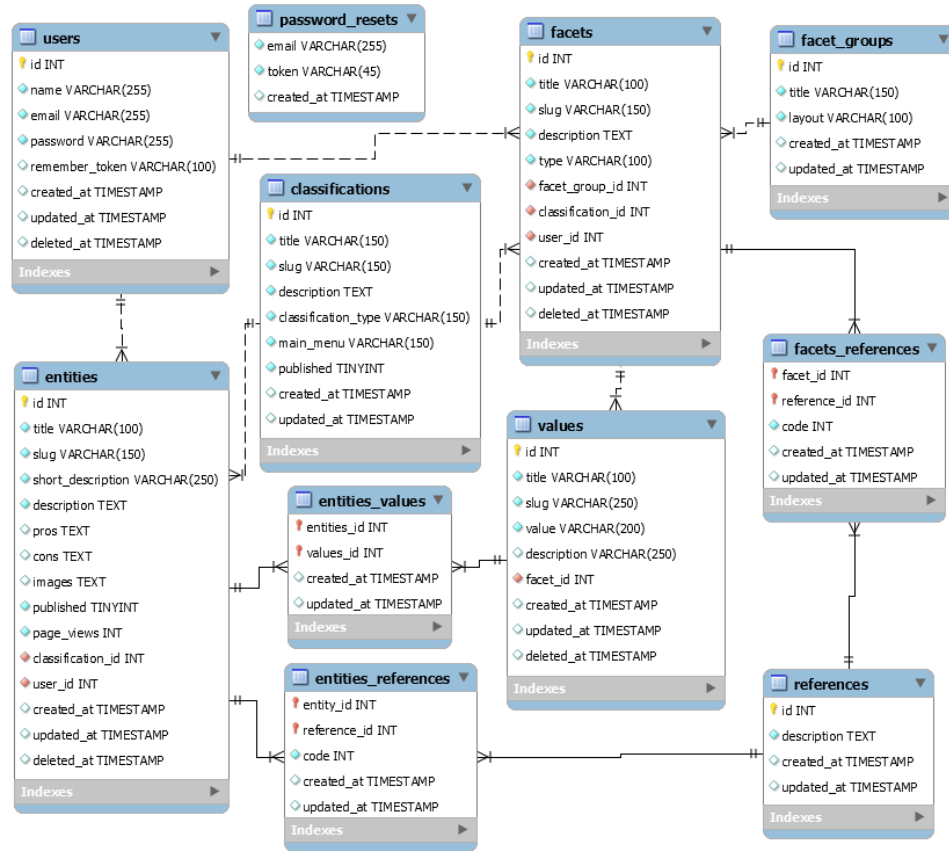
5.3 MODELAGEM DE DADOS

A modelagem de dados é uma técnica para organizar e documentar como os dados do sistemas serão representados. Um dos diagramas utilizados para este fim é chamado Modelo Entidade-Relacionamento, no qual os dados são representados por entidades e o relacionamento entre elas (LI; CHEN, 2009). A **Figura 14** apresenta o modelo adotado para o guia proposto no presente trabalho. A modelagem foi desenvolvida com base na proposta apresentada por Denton em seu artigo intitulado *How to make a faceted classification and put it on the web* (DENTON, 2003).

5.4 APLICAÇÃO

A aplicação está separada em duas camadas. A primeira é apenas de visualização, onde é apresentado o guia. Esta camada foi desenvolvida utilizando HTML, CSS e Ja-

Figura 14 – Modelo Entidade-Relacionamento



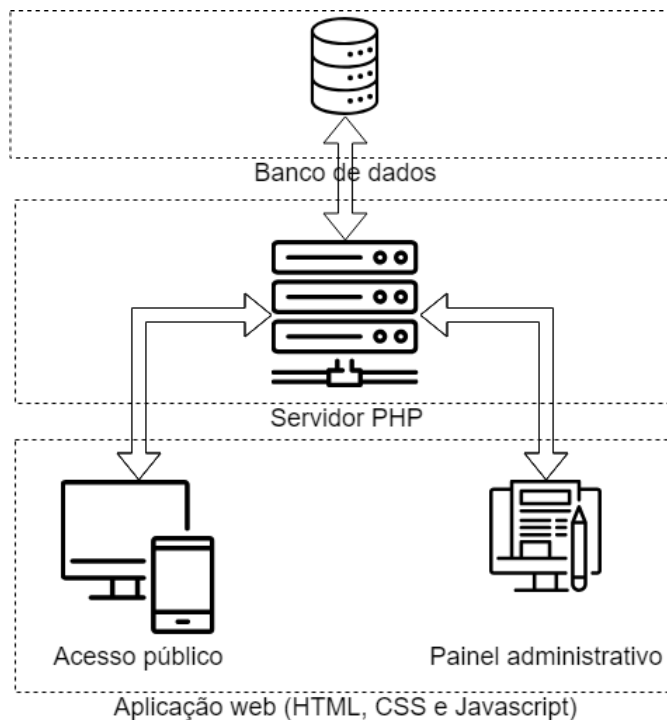
Fonte: Elaboração do autor, 2018

vascript por meio de um framework chamado VueJS (VUEJS, 2018). A segunda camada consiste na estrutura do servidor da aplicação para responder as requisições efetuadas pelos usuários. Neste caso, além das linguagens de programação utilizadas na camada de visualização, foi utilizado o PHP (THE PHP GROUP, 2018) como linguagem principal com o framework Laravel (LARAVEL, 2018) e o banco de dados MySQL (ORACLE CORP., 2018) para persistência dos dados. O detalhamento de cada uma das tecnologias utilizadas é apresentado nas **Seções 5.4.1 e 5.4.2**. A **Figura 15** apresenta estruturalmente como se dá a comunicação entre as tecnologias utilizadas.

5.4.1 Servidor

Esta camada foi construída principalmente utilizando a linguagem de programação PHP, através de um framework chamado Laravel, gratuito e de código aberto que utiliza MVC (Modelo, Visão e Controle) como padrão de projeto para desenvolvimento de aplicações web (PUTRA; KANIGORO et al., 2015). Até o momento do desenvolvimento da aplicação, a versão estável deste framework era a 5.6. Foram utilizadas todas práticas pre-

Figura 15 – Comunicação entre camadas da aplicação



Fonte: Elaboração do autor, 2018

sentes na documentação oficial para facilitar a configuração, manutenção e compreensão da ferramenta. Seus principais objetivos neste projeto são:

- Fornecer uma rota de entrada para o guia.
- Fornecer uma API de consumo para a aplicação cliente.
- Fornecer um painel administrativo para gestão do conteúdo apresentado.
(<http://retraining.inf.ufsc.br/guia/admin>)

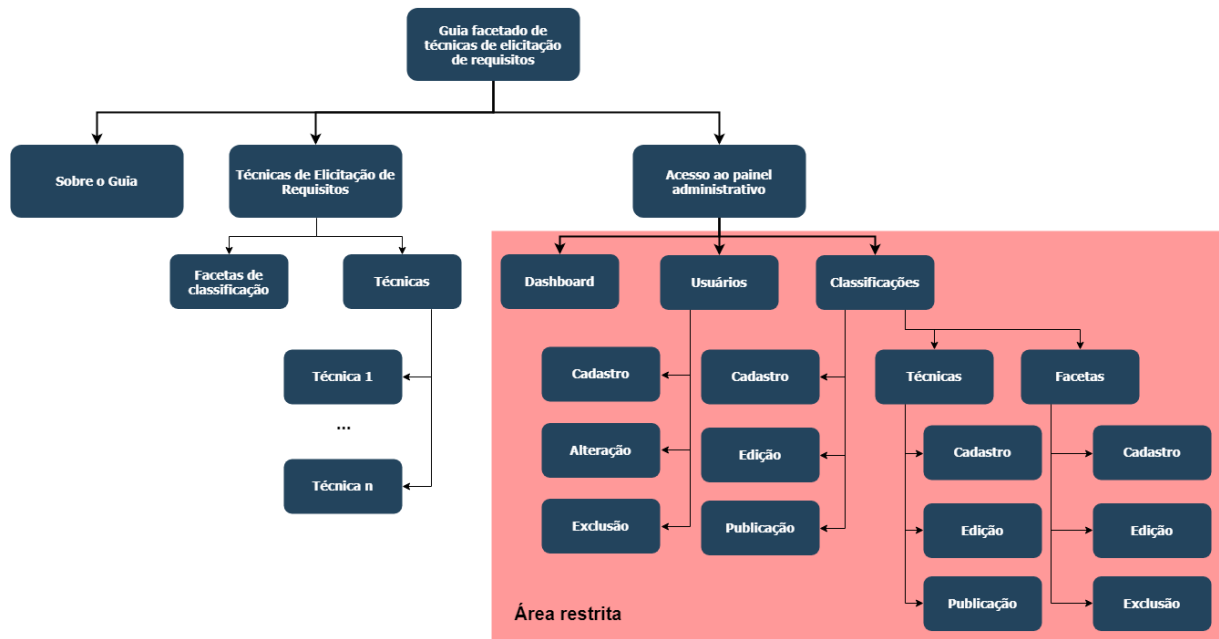
Todos os passos para configuração da plataforma, assim como uma explicação mais detalhada de cada item desenvolvido, estão expostos em um arquivo intitulado RE-ADME.md na raiz do diretório do projeto entregue anexo ao presente trabalho.

5.4.2 Cliente

No lado do cliente, além das tecnologias mais conhecidas para desenvolvimento de sistemas web, como HTML, CSS e Javascript, foi utilizado um framework chamado VueJS. VueJS é um framework progressivo para desenvolvimento de interfaces para usuário, desenvolvido em javascript, gratuito e de código fonte aberto. Recebe o título de progressivo pois tem o foco na facilidade de integrar com outras bibliotecas ou projetos

existentes (VUEJS, 2018). Seu objetivo é fornecer uma interface de acesso ao guia, por onde os usuários consultam as informações disponibilizadas. Sua configuração também está detalhada no repositório do código do projeto. O link de acesso a este ambiente é <http://retraining.inf.ufsc.br/guia/app>. A **Figura 16** apresenta o mapa de acesso ao guia de forma completa.

Figura 16 – Mapa de acesso ao guia



Fonte: Elaboração do autor, 2018

6 AVALIAÇÃO

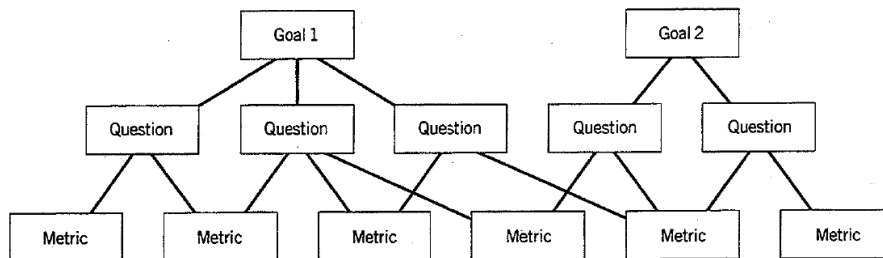
Com o intuito de avaliar as informações coletadas e o Guia desenvolvido, agregando ainda mais valor para o presente trabalho, faz-se necessária a utilização de um método estabelecido para guiar a avaliação a ser realizada. A opção adotada neste caso foi a abordagem GQM (Goal-Question-Metric), um mecanismo para planejamento e definição dos objetivos a serem medidos e avaliados (CALDIERA; ROMBACH, 1994).

O modelo resultante possui 3 níveis (CALDIERA; ROMBACH, 1994):

- Goal: Definição de uma meta para determinado objeto a ser avaliado no contexto do projeto.
- Question: Uma série de perguntas usadas para caracterizar uma meta específica de um objetivo que será realizado.
- Metric: Uma série de indicadores associados a cada pergunta realizada.

A **Figura 17** apresenta a relação entre entes níveis.

Figura 17 – Estrutura hierárquica do modelo GQM

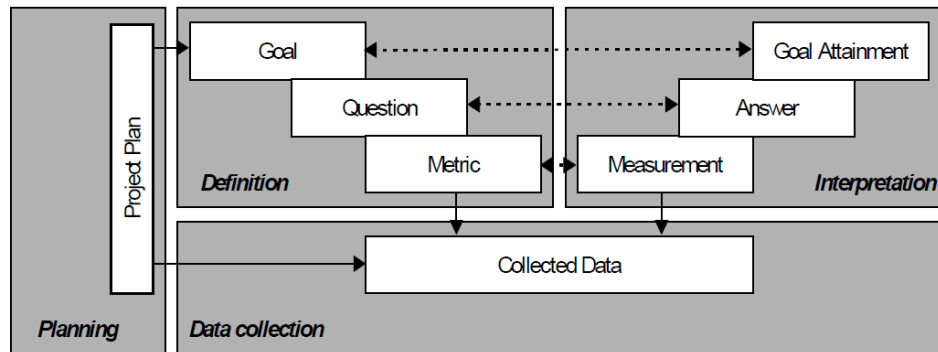


Fonte: (CALDIERA; ROMBACH, 1994)

Solingen e Berghout (1999) divide o método GQM em 4 fases (**Figura 18**):

- Planejamento: Criação de um plano de projeto onde são definidas as diretrizes para a aplicação da medição.
- Definição: Definição e documentação do programa de medição (objetivos, perguntas, métricas e hipóteses).
- Coleta de dados: Fase de execução do plano de projeto, onde os dados são coletados.
- Interpretação: Processamento dos dados coletados em relação às métricas definidas no plano de projeto, fornecendo respostas para as perguntas criadas.

Figura 18 – As quatro fases do método GQM



Fonte: (SOLINGEN; BERGHOUT, 1999)

6.1 PLANEJAMENTO

A aplicação do modelo tem como principal objetivo avaliar o conteúdo produzido, a organização das informações e o guia onde a informação é apresentada.

6.2 DEFINIÇÃO

O primeiro objetivo diz respeito a qualidade e organização do conteúdo produzido. A **Tabela 14** formaliza este objetivo.

Tabela 14 – Objetivo de medição 1 - Avaliação do conteúdo

Analisar o	guia de técnicas de elicitação de requisitos
Com o propósito de	avaliar a facilidade de entendimento
Com respeito a	apresentação de novas técnicas e compreensão das mesmas
Do ponto de vista	de profissionais e acadêmicos
No contexto da	elicitação de requisitos

Fonte: Elaboração do autor, 2018

Para atender este objetivo é necessário responder as questões apresentadas nas **Tabelas 15 e 17** através de suas respectivas métricas. Esta derivação tem dois pontos principais, o primeiro está na comparação das perguntas associadas ao caso de exemplo apresentado e a segunda está na coleta da opinião dos entrevistados com relação ao nível de compreensão e adequação da ferramenta utilizada.

Tabela 15 – Questão 1 para o Objetivo 1 e suas respectivas métricas

1.1 Questão: Quão efetivo é o guia em sugerir novas técnicas
M1.1.1) Quais técnicas para elicitação (identificação) de requisitos você conhece?
M1.1.2) Dado um caso de exemplo (Tabela 16), quais técnicas você utilizaria para a elicitação de requisitos?
M1.1.3) Após a pesquisa no guia, quais técnicas você poderia utilizar para o caso de exemplo avaliado anteriormente (Tabela 16)?

Fonte: Elaboração do autor, 2018

Tabela 16 – Caso de exemplo para utilização na coleta de dados

Caso de exemplo:

Supondo que você trabalhe como analista em uma empresa bem estabelecida no mercado que desenvolve software para o ramo de supermercados*. A empresa acaba de aprovar um novo projeto de software para lojas de pequeno porte. As regras de negócio são bem similares ao que a empresa já conhece, facilitando no entendimento do problema**. Devido a alta temporada, os stakeholders acabam passando muito tempo em viagens negociando com fornecedores e não conseguem se reunir para definir os objetivos do projeto***.

* Contextualização da situação.

** Condição que indica de que o analista e a empresa já possuem conhecimento prévio quanto ao problema abordado.

*** Condição que indica que os stakeholders serão consultados indiretamente, de forma individual e nem sempre estarão perto do analista.

As características destacadas (*, **, ***) devem induzir o entrevistado a escolher entre as técnicas Questionários, Laddering ou Repertory grids devido a sua classificação.

Fonte: Elaboração do autor, 2018

Tabela 17 – Questão 2 para o Objetivo 1 e suas respectivas métricas

1.2 Questão: Qual o grau de compreensão das técnicas apresentadas pelo Guia de elicitação de requisitos
M1.1.1) O conteúdo apresentado no guia é de fácil compreensão.
M1.1.2) O conteúdo do guia é adequado (suficiente, claro e correto) em relação as técnicas acessadas.
Para estas métricas deve ser utilizada a escala Likert apresentada na Tabela 18

Fonte: Elaboração do autor, 2018

Tabela 18 – Escala likert utilizada para medições da questão 1.2

Fórmula	Interpretação			
X=A, Onde A=	(1) Discordo totalmente	(2) Discordo parcialmente	(3) Concordo parcialmente	(4) Concordo totalmente

Fonte: Elaboração do autor, 2018

O segundo objetivo diz respeito ao Guia, mais especificamente quanto a usabilidade da ferramenta. Este objetivo está formalizado na **Tabela 19**.

Tabela 19 – Objetivo de medição 2 - Análise da ferramenta

Analisar o	guia de técnicas de elicitação de requisitos
Com o propósito de	avaliar a facilidade de utilização da ferramenta
Com respeito a	navegação e disposição das informações
Do ponto de vista	de usuário
No contexto da	aplicação

Fonte: Elaboração do autor, 2018

Os aspectos de usabilidade foram avaliados por meio da escala SUS (System Usability Scale), um método simples que procura mensurar a usabilidade de sistemas através de algumas perguntas, respondidas através da escala Likert (de 1 a 5, onde 1 significa discordo completamente e 5 significa concordo completamente), que resultam em uma pontuação final (MCLELLAN; MUDDIMER; PERES, 2012). A aplicação da escala SUS resulta em um indicador que corresponde ao nível de usabilidade da ferramenta avaliada. Este indicador é utilizado como métrica para os objetivos levantados. A questão associada à escala SUS e suas respectivas métricas são apresentadas na **Tabela 20**.

Tabela 20 – Questão 1 para o Objetivo 2 e suas respectivas métricas

2.1 Questão: Qual a pontuação na escala SUS obtida pela ferramenta
M2.1.1) Eu acho que gostaria de usar esse sistema com frequência.
M2.1.2) Eu acho o sistema desnecessariamente complexo.
M2.1.3) Eu achei o sistema fácil de usar.
M2.1.4) Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
M2.1.5) Eu acho que as várias funções do sistema estão muito bem integradas.
M2.1.6) Eu acho que o sistema apresenta muita inconsistência.
M2.1.7) Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
M2.1.8) Eu achei o sistema atrapalhado de usar.
M2.1.9) Eu me senti confiante ao usar o sistema.
M2.1.10) Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Fonte: Elaboração do autor, 2018

Para a identificação dos entrevistados e de seus respectivos perfis, serão aplicadas as perguntas:

- Qual seu nome?

- Em qual empresa você trabalha?
- Você possui formação superior completa? (Sim ou Não)
- Você possui um curso superior em andamento? (Sim ou Não)
- Qual curso que você fez ou está fazendo?
- Você já cursou disciplinas associadas a Engenharia de requisitos? (Sim ou Não)
- Você já executou atividades relacionadas a Engenharia de requisitos profissionalmente (licitação, análise, documentação, validação)? (Sim ou Não)
- Qual papel você frequentemente assume nos projetos? (Analista de negócios, Engenheiro de requisitos/Analista de sistemas, Gerente de projeto/Líder de projeto, Testador/Gerente de testes, Projetista, Desenvolvedor, Não estou trabalhando ou Outro)
- Quanto tempo você teve ou tem com a atividade de levantamento/análise de requisitos? (menos de 1 ano, de 1 a 3 anos ou mais de 3 anos)

6.3 COLETA DE DADOS

A coleta de dados foi feita através de entrevistas estruturadas (**Apêndice D**). Segundo Gill et al. (2008) neste método um questionário é aplicado verbalmente através de perguntas predeterminadas, tornando a execução rápida e de fácil administração. A entrevista era composta por 24 perguntas, diversificadas entre identificação, conhecimento acerca do tema, avaliação do conteúdo apresentado e avaliação da ferramenta. Os entrevistados foram classificados em dois tipos conforme suas respostas na etapa do questionário responsável pela sua identificação (**Tabela 21**).

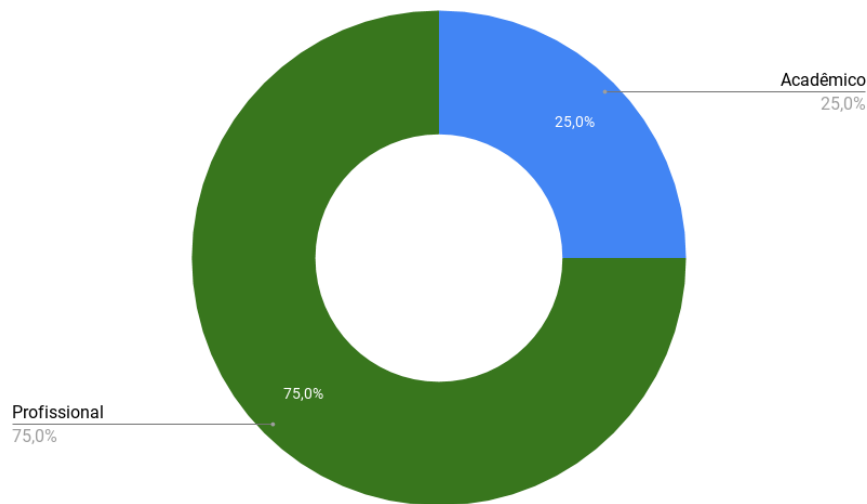
Tabela 21 – Classificação dos participantes segundo suas informações

Tipo	Características
Acadêmico	Curso concluído ou em andamento na área de TI Ter cursado uma disciplina associada a engenharia de requisitos Ter menos de 1 ano de experiência com a atividade de levantamento/análise de requisitos
Profissional	Ter 1 ano ou mais de experiência com a atividade de levantamento/análise de requisitos

Fonte: Elaboração do autor, 2018

Foram selecionadas pessoas próximas ao autor, do meio acadêmico e do meio profissional, com ou sem experiência na atividade de elicitação de requisitos e que estão inseridas no meio de tecnologia da informação. A quantidade de entrevistados não foi estipulada inicialmente, o único critério para esta questão é que a coleta deveria ter uma quantidade balanceada em relação ao tempo de experiência na atividade de elicitação de requisitos. Ao todo foram entrevistadas 16 pessoas, 4 categorizados como acadêmicos e 12 categorizados como profissionais conforme definidos na **Tabela 21** e apresentados na **Figura 19**.

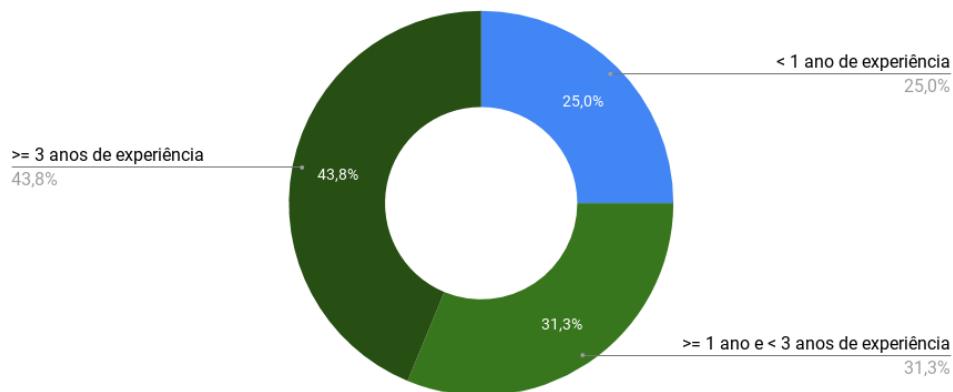
Figura 19 – Total de entrevistados por categoria



Fonte: Elaboração do autor, 2018

A principal característica que implica na categorização dos entrevistados é sua experiência quanto ao tema estudado, como consta na **Figura 20**.

Figura 20 – Total de entrevistados por experiência



Fonte: Elaboração do autor, 2018

Os dados coletados são apresentados nas **Tabelas 22, 23 e 24**. A primeira coluna representa cada um dos entrevistados e as demais colunas são suas respostas para as perguntas apresentadas conforme consta na **Tabela 26 (Apêndice D)**.

Tabela 22 – Identificação dos entrevistados

Entrevistado	Q3	Q4	Q5	Q6	Q7	Q8	Q9
E1	Não	Sim	Sistemas de informação	Sim	Sim	Analista de negócios	Experiente (1-3 anos de experiência)
E2	Não	Sim	Sistemas de informação	Sim	Sim	Desenvolvedor	Novato (menos de 1 ano de experiência)
E3	Não	Sim	Sistemas de Informação	Sim	Sim	Desenvolvedor	Novato (menos de 1 ano de experiência)
E4	Não	Sim	Sistemas de Informação	Sim	Não	Desenvolvedor	Novato (menos de 1 ano de experiência)
E5	Não	Sim	Sistemas de Informação	Sim	Sim	Desenvolvedor	Novato (menos de 1 ano de experiência)
E6	Não	Sim	Sistemas de Informação	Sim	Não	Desenvolvedor	Expert (mais de 3 anos de experiência)
E7	Não	Sim	Sistemas de Informação	Sim	Sim	Desenvolvedor	Novato (menos de 1 ano de experiência)
E8	Não	Sim	Sistemas de Informação	Sim	Sim	Desenvolvedor	Novato (menos de 1 ano de experiência)
E9	Sim	Não	Bacharel em sistemas de informação, Governança em TI e Engenharia de Software	Sim	Sim	Gerente de projeto/Líder de projeto	Expert (mais de 3 anos de experiência)
E10	Sim	Não	Ciências da Computação, Administração e Pedagogia	Sim	Sim	Desenvolvedor	Expert (mais de 3 anos de experiência)
E11	Sim	Não	Ciências da Computação	Sim	Sim	Gerente de projeto/Líder de projeto	Expert (mais de 3 anos de experiência)
E12	Sim	Não	Ciências da Computação	Sim	Sim	Projetista	Expert (mais de 3 anos de experiência)
E13	Sim	Não	Psicologia, Análise e Desenvolvimento de Sistemas	Sim	Sim	Product Owner	Experiente (1-3 anos de experiência)
E14	Sim	Não	Ciência da Computação	Não	Sim	Engenheiro de requisitos/Analista de sistemas	Expert (mais de 3 anos de experiência)
E15	Sim	Não	Sistemas de Informação	Sim	Sim	Analista de negócios	Expert (mais de 3 anos de experiência)
E16	Não	Não	Técnico em Informática	Não	Sim	Gerente de projeto/Líder de projeto	Expert (mais de 3 anos de experiência)

Fonte: Elaboração do autor, 2018

Tabela 23 – Respostas relacionadas às questões acerca do tema proposto

Entrevistado	Q10	Q11	Q12
E1	Entrevistas, casos de uso, prototipação	Passo 1: Elaboração de um documento com as funcionalidades e regras similares ao outro sistema. Passo 2: Prototipação	Casos de uso, prototipação, cenários
E2	Scrum, Entrevistas, Imersão no ambiente de trabalho	Identificação de stakeholders a nível operacional (líderes de operação por exemplo), ligações, e-mails.	Questionários, Observação direta
E3	Entrevista	Entrevista. Envolvimento com o ambiente das lojas de pequeno porte para entender os processos comuns na mesma.	Observação direta; Entrevista; Brainstorming; Cenários.
E4	Entrevista	Entrevista por video chamada	Questionário
E5	Entrevista.	Utilizaria ferramentas de comunicação assíncrona para a comunicação e faria a elicitação de requisitos remotamente.	Utilizaria a técnica "Questionários".
E6	Entrevista, história	Entrevista individual com cada stakeholder	Questionários
E7	Reunião com cliente para entender o negócio	Reuniões esporádicas com clientes para confirmar os requisitos já conhecidos pela empresa.	Questionários
E8	Store uma das etapas do processo Scrum.	Trocas de e-mails, vídeo conferencia.	Entrevista, Brainstorming
E9	Entrevista com o cliente e prototipação	Prototipação	Observação direta, Análise de documentos existentes, questionários
E10	Nenhuma	Observação dos softwares legados e coleta de informações específicas com os stakeholders	Laddering e questionarios
E11	Entrevistas, Análise de documentos	Análise de documentos	Análise de Documentos e Cenários
E12	Entrevista	Entrevista, Berchmarking,	Entrevistas, Questionários, Repertory Grids
E13	Entrevista	Revisão do fluxo já utilizado e das principais funções do sistema; Conversa com pessoas das áreas que possivelmente utilizariam o sistema para entender as necessidades;	Entrevista, Análise de documento, Questionário, Observação direta
E14	Histórias ou entrevistas, questionários, prototipagem e Brainstorming	Questionários e prototipagem	Questionário e prototipagem
E15	Técnicas de design thinking, levantamento de personas, mapa de fluxo, jobs to be done, entrevistas, pesquisa	Mapa de jornada de usuários, jobs to be done	Cenários, Entrevista, Análise de documentos
E16	Reunião com o cliente, acompanhamento de processo, estudo de casos similares	Levantamento de requisitos dos casos anteriores	Questionários, Repertory Grids

Fonte: Elaboração do autor, 2018

Tabela 24 – Avaliação do conteúdo produzido e da ferramenta

Entrevistado	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24
E1	4	4	5	1	5	1	5	1	5	1	5	4
E2	4	4	5	2	5	2	5	1	4	1	4	1
E3	4	4	5	1	4	1	5	1	4	1	5	1
E4	4	4	4	1	5	1	5	1	5	2	5	1
E5	4	4	4	1	4	1	5	1	5	1	5	1
E6	4	4	3	1	5	1	5	1	5	1	5	1
E7	4	4	4	1	5	1	5	1	5	1	5	1
E8	3	3	3	3	4	4	4	2	4	2	3	4
E9	4	4	4	2	4	1	5	1	5	1	5	1
E10	4	4	5	1	5	1	5	1	5	1	5	5
E11	3	2	4	2	3	2	3	1	5	4	3	2
E12	4	4	5	1	5	1	5	1	5	1	5	1
E13	4	4	5	1	5	1	5	1	5	1	4	1
E14	4	4	5	1	5	1	5	1	5	1	5	1
E15	4	4	5	2	4	1	4	1	4	1	4	1
E16	4	4	4	1	4	2	5	1	5	1	5	1

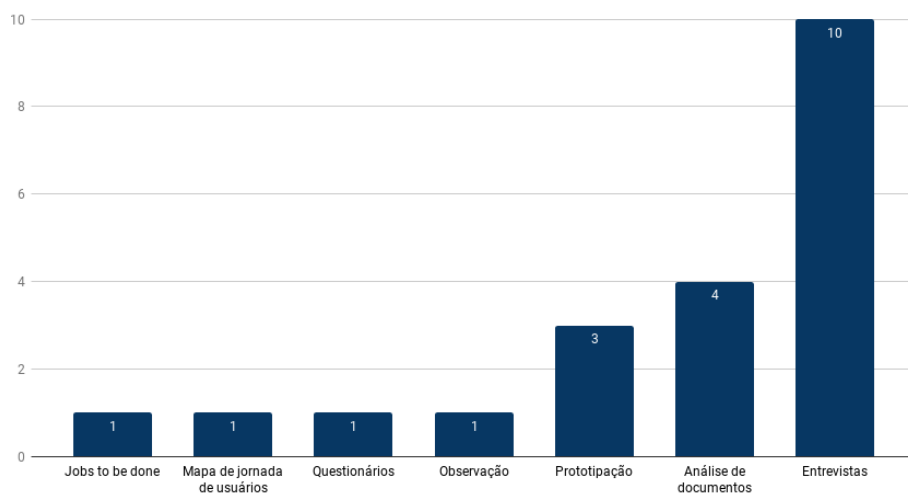
Fonte: Elaboração do autor, 2018

6.4 INTERPRETAÇÃO

Para responder sobre a efetividade do guia em sugerir novas técnicas (**Questão 1.1**) foram feitas perguntas acerca do tema, antes e depois da apresentação do guia. Primeiramente foram questionadas quais técnicas conhecidas pelo entrevistado. As respostas coletadas não apresentaram um número alto de técnicas e em alguns casos, os termos citados não eram adequados à etapa de Elicitação de Requisitos.

Em um segundo momento, a partir da apresentação de um caso de exemplo, foi perguntado aos entrevistados quais técnicas mais adequadas para a elicitação de requisitos, considerando as condições associadas ao problema em questão. A interpretação das características apresentadas também era parte do processo. Na maioria dos casos, as respostas não foram adequadas ao problema apresentado. Apenas 1 participante citou uma técnica esperada como uma opção válida (**Figura 21**).

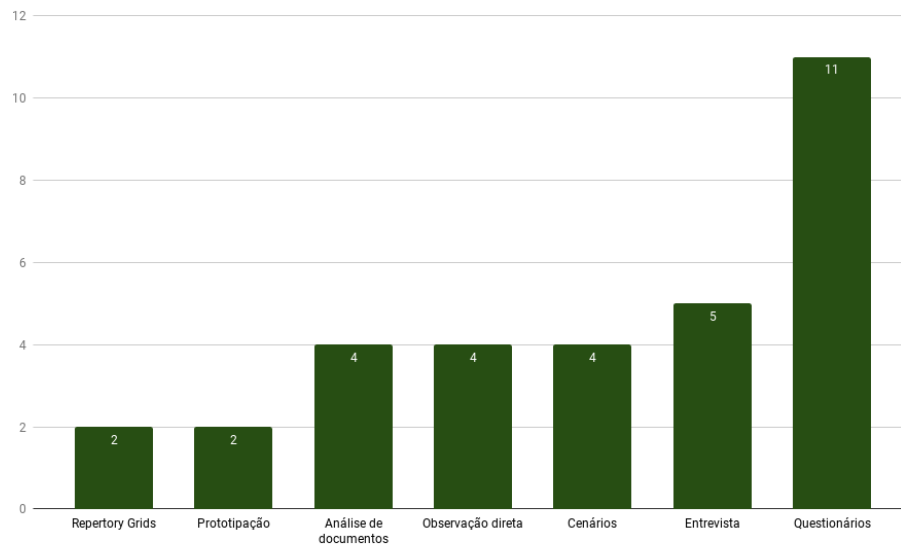
Figura 21 – Técnicas citadas antes da apresentação do guia



Fonte: Elaboração do autor, 2018

Após a apresentação do guia, o caso de exemplo foi reapresentado e novamente foi questionado quais técnicas poderiam ser utilizadas. Alguns entrevistados ainda apresentaram técnicas não muito adequadas, mas desta vez a maioria apresentou as respostas esperadas para o problema (**Figura 22**).

Figura 22 – Técnicas citadas depois da apresentação do guia

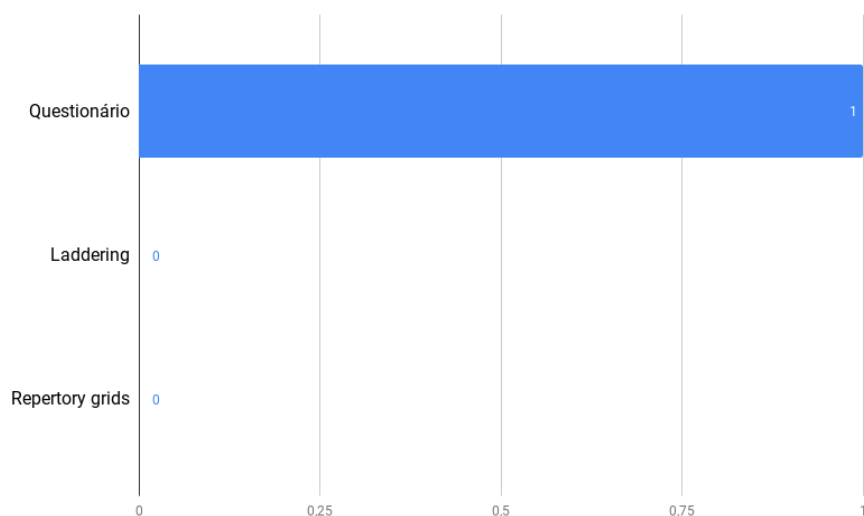


Fonte: Elaboração do autor, 2018

Cabe destacar também que, ao comparar as **Figuras 21 e 22**, é possível perceber que cerca de 69% dos participantes mudaram de ideia após a utilização do guia, indicando uma certa influência do guia sobre suas respostas.

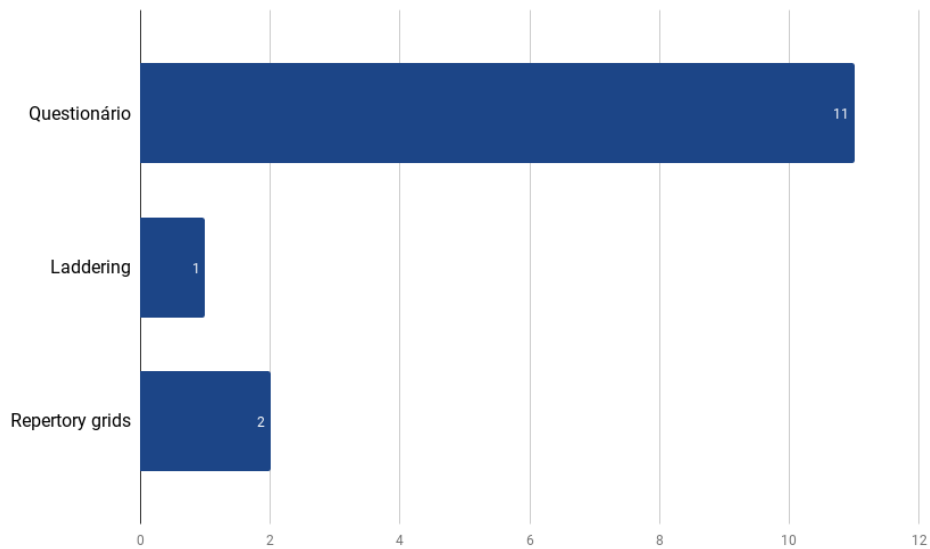
Na maioria dos casos (69%), os entrevistados informaram uma ou mais técnicas dentre as possíveis opções para o caso de exemplo apresentado. A quantidade de respostas adequadas aos resultados esperados, assim como uma comparação entre o momento anterior e posterior da apresentação do guia podem ser vistos na **Figuras 23 e 24**.

Figura 23 – Total de resultados esperados antes da apresentação do guia



Fonte: Elaboração do autor, 2018

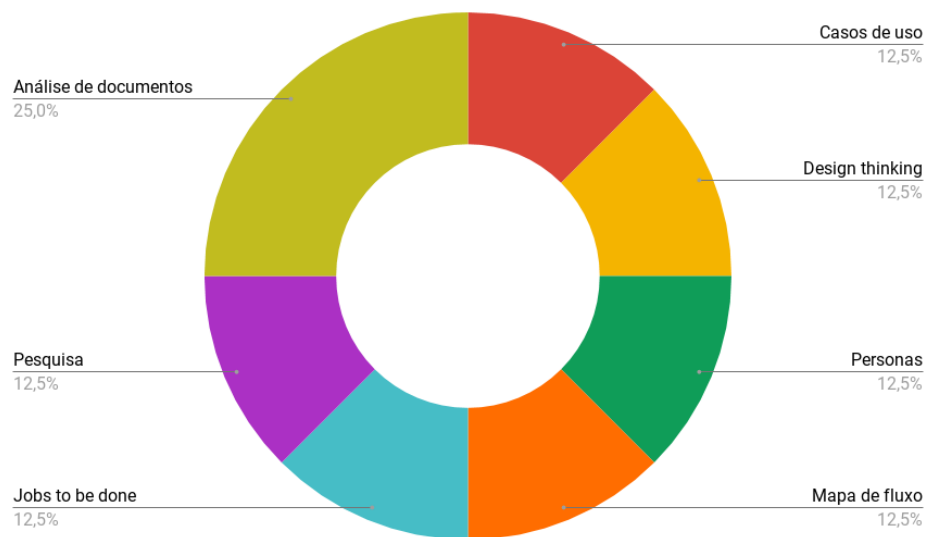
Figura 24 – Total de resultados esperados após da apresentação do guia



Fonte: Elaboração do autor, 2018

A diferença entre os dois cenários é muito clara, pode-se perceber que a ferramenta de fato auxiliou os participantes a chegarem em uma das respostas esperadas para o caso de exemplo. Por fim, é possível perceber também que metade dos participantes citaram em suas respostas, técnicas que ainda não estão presentes no guia. Na **Figura 25** são apresentadas as técnicas citadas pelos entrevistados que não constam no guia. *Análise de documentos* se destaca como a técnica que foi citada mais vezes e não foi classificada.

Figura 25 – Técnicas citadas que não estão presentes no guia



Fonte: Elaboração do autor, 2018

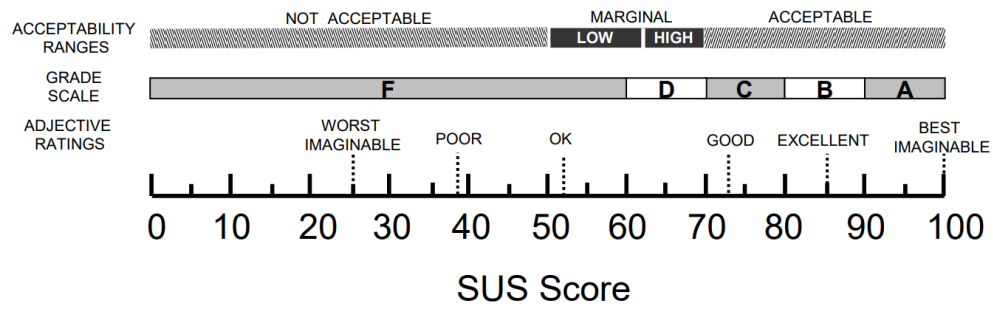
Complementando a avaliação do conteúdo apresentado, foram efetuadas perguntas acerca da qualidade e adequação do conteúdo produzido e apresentado no guia (**Questão 1.2**). Foram feitas perguntas utilizando uma escala Likert, onde os entrevistados poderiam optar entre, discordo totalmente, discordo parcialmente, concordo parcialmente e concordo totalmente para as afirmações associadas à compreensão do conteúdo e adequação do conteúdo em relação às técnicas acessadas. Para a primeira métrica desta questão, foi levantado que 88% dos participantes concordaram totalmente que a ferramenta possui um conteúdo de fácil compreensão, os demais entrevistados apenas concordaram parcialmente com essa afirmação. A proporção de entrevistados que concordaram totalmente com a afirmação de que o conteúdo é adequado para as técnicas acessadas se manteve em 88%, sendo que 6% concordaram parcialmente e outros 6% discordaram parcialmente.

Antes de entrar na análise das estatísticas levantadas, é interessante destacar que, na maioria das vezes os entrevistados não tinham certeza ou não sabiam a nomenclatura das técnicas. Para estes casos os participantes foram incentivados a informar as técnicas da maneira que tinham conhecimento. Algumas respostas associadas às técnicas eram inadequadas ao tema, ou seja, não se tratavam de técnicas de eliciação de requisitos.

Como resultado do primeiro objetivo da avaliação, consideramos que o guia auxiliou os entrevistados apresentando técnicas que eles não conheciam ou não lembravam. Com o acesso à plataforma foi possível guiar a escolha de quais técnicas mais adequadas para utilização em determinado cenário. Os filtros também foram essenciais para encontrar as técnicas com base nas características do caso de exemplo apresentado, justificando a utilização da classificação facetada neste trabalho. O conteúdo produzido foi avaliado positivamente pela maioria dos participantes, tanto para as afirmações relacionadas a compreensão do conteúdo quanto para as afirmações associadas à adequação dos conteúdos de cada uma das técnicas apresentadas.

Quanto ao segundo objetivo de avaliação (**Tabela 19**), foram analisados os aspectos de usabilidade associados à ferramenta desenvolvida. Para responder a pergunta relacionada a usabilidade, optou-se em utilizar a escala SUS, conforme apresentado na **Seção 6.2**. Esta escala resulta em um valor que indica o nível de usabilidade de uma aplicação, conforme apresentado na (**Figura 26**). Este valor é calculado conforme cada uma das respostas dos participantes, de onde é extraída uma média. Para o guia apresentado, a escala SUS resultante foi de 90. Este valor indica que a plataforma possui um nível de usabilidade aceitável dentro a escala utilizada.

Figura 26 – Escalas de avaliação conforme a pontuação SUS



Fonte: (BANGOR; KORTUM; MILLER, 2009)

7 CONSIDERAÇÕES FINAIS

A pesquisa demonstrou que existem inúmeras técnicas de elicitação de requisitos, foram encontradas cerca de 96 técnicas conforme os critérios aplicados na revisão sistemática. Foram encontradas desde técnicas bem estabelecidas no mercado até técnicas que estão em fase de experimentação. Estes fatores validam as informações citadas por outros autores e citadas no início deste trabalho (**Capítulo 1**) referente a existência de um número considerável de técnicas e na viabilização de um guia condensando suas informações.

A utilização da classificação facetada neste trabalho demonstrou o quão simples e útil pode ser este método. A seleção das características a serem tratadas como facetas se mostrou como a parte mais importante de todas. Uma falha nesta etapa pode acarretar em classificações incoerentes ou que não apresentam um resultado expressivo quanto ao agrupamento de seus elementos. Foram classificadas 12 técnicas de elicitação de requisitos, um número relativamente baixo se comparado às 96 técnicas encontradas no estudo, mas que já demonstraram resultados interessantes para as buscas efetuadas.

A avaliação da ferramenta apresentou resultados positivos tanto para os conteúdos produzidos quanto para a ferramenta desenvolvida, demonstrando que as decisões tomadas foram assertivas no decorrer da execução. Uma parte significativa dos entrevistados (50%) acabaram citando algumas técnicas que não estavam presentes no guia. Com isso, entende-se que é interessante que seu conteúdo seja atualizado com a inclusão de outras técnicas de elicitação de requisitos, com o objetivo de manter uma base mais completa possível. Além das estatísticas apresentadas na **Seção 6.4**, é importante mencionar que grande parte dos entrevistados apresentaram interesse que a ferramenta fosse divulgada quando publicada oficialmente.

Por fim, é importante destacar que, como acadêmico, este trabalho proporcionou grande aprendizado durante sua execução. Além do conhecimento adquirido sobre o tema proposto, ou seja, técnicas de elicitação de requisitos, classificação facetada e desenvolvimento do guia, o trabalho proporcionou um aprendizado acerca dos processos de pesquisa científica e seus derivados.

7.1 TRABALHOS FUTUROS

A partir dos resultados alcançados, ficam como sugestões de trabalhos a serem desenvolvidos:

- Coleta e classificação das demais técnicas encontradas a fim de completar o guia e fornecer uma base centralizada de técnicas conhecidas, enriquecendo ainda mais a plataforma e fornecendo insumos para os filtros realizados.
- Classificação de técnicas utilizadas em outras etapas da engenharia de requisitos, uma vez que para cada etapa deste processo existe uma variedade de técnicas que podem ser utilizadas.
- Revisão e avaliação de facetas para a composição de uma classificação mais completa e robusta, com características mais alinhadas com a necessidade dos usuários que procuram combinações para o seu cenário.
- Mapeamento das relações entre as técnicas de elicitação e apresentação no guia, uma vez que as técnicas podem ser utilizadas com objetivos diferentes, uma relação poderia ajudar a achar uma combinação ou o descarte da utilização de métodos similares.

REFERÊNCIAS

- ABBASI, M. A. et al. Assessment of requirement elicitation tools and techniques by various parameters. *Software Engineering*, v. 3, n. 2, p. 7–11, 2015.
- ALEGRE, U.; AUGUSTO, J. C.; CLARK, T. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, Elsevier, v. 117, p. 55–83, 2016. ISSN 0164-1212.
- ARAÚJO, C. A. Á. Fundamentos teóricos da classificação. *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, Universidade Federal de Santa Catarina, n. 22, p. 117–140, 2006. ISSN 1518-2924. <<https://periodicos.ufsc.br/index.php/eb/article/view/1518-2924.2006v11n22p117>>.
- ARIF, Q. K. Shams-ul; GAHYUR, S. Requirements engineering processes, tools/technologies, & methodologies. *International Journal of Reviews in Computing*, p. 41–56, 2009.
- BANGOR, A.; KORTUM, P.; MILLER, J. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, Usability Professionals' Association, v. 4, n. 3, p. 114–123, 2009.
- BARBOSA, A. P. Classificações facetadas. *Ciência da Informação*, v. 1, n. 2, 1972. ISSN 1518-8353. <<http://revista.ibict.br/ciinf/article/view/10>>.
- BATISTA, E. A.; CARVALHO, A. M. B. R. Uma taxonomia facetada para técnicas de elicitação de requisitos. In: *WER*. [S.l.: s.n.], 2003.
- BELGAMO, A.; EDUARDO, L.; MARTINS, L. E. Estudo comparativo sobre as técnicas de elicitação de requisitos do software. 01 2008.
- BESROUR, S.; RAHIM, L. B. A.; DOMINIC, P. Assessment and evaluation of requirements elicitation techniques using analysis determination requirements framework. In: IEEE. *Computer and Information Sciences (ICCOINS), 2014 International Conference on*. [S.l.], 2014. p. 1–6.
- BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, Elsevier, v. 80, n. 4, p. 571–583, 2007.
- CALDIERA, V. R. B.-G.; ROMBACH, H. D. Goal question metric paradigm. *Encyclopedia of software engineering*, v. 1, p. 528–532, 1994.
- CAMPOS, M. A. *Linguagem documentária: teorias que fundamentam sua elaboração*. Ed. UFF, 2001. ISBN 9788522803194. <<https://books.google.com.br/books?id=xgABnwEACAAJ>>.
- CARRIZO, D.; DIESTE, O.; JURISTO, N. Systematizing requirements elicitation technique selection. *Information and Software Technology*, Elsevier, v. 56, n. 6, p. 644–669, 2014. ISSN 0950-5849.

- DAVIS, A. et al. Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. In: IEEE. *Requirements Engineering, 14th IEEE International Conference*. IEEE Computer Society, 2006. (RE '06), p. 179–188. ISBN 0-7695-2555-5. <<https://doi.org/10.1109/RE.2006.17>>.
- DAVIS, A. M. Operational prototyping: A new development approach. *IEEE software*, IEEE, v. 9, n. 5, p. 70–78, 1992.
- DAVIS, G. B. Strategies for information requirements determination. *IBM systems journal*, IBM, v. 21, n. 1, p. 4–30, 1982.
- DENTON, W. How to make a faceted classification and put it on the web. See <http://www.miskatonic.org/library/facet-web-howto.html>, 2003.
- DIESTE, O.; JURISTO, N. Systematic review and aggregation of empirical studies on elicitation techniques. *IEEE Transactions on Software Engineering*, IEEE, v. 37, n. 2, p. 283–304, March 2011. ISSN 0098-5589.
- DIESTE, O.; JURISTO, N.; SHULL, F. Understanding the customer: what do we know about requirements elicitation? *IEEE Software*, IEEE, v. 25, n. 2, p. 11–13, March 2008. ISSN 0740-7459.
- DIRGAHAYU, T.; SETIANI, N.; ZUKHRI, Z. Information requirements engineering for specific content management systems. In: IEEE. *Open Systems (ICOS), 2014 IEEE Conference on*. [S.l.], 2014. p. 54–59.
- DUARTE, D. et al. Collaborative requirements elicitation with visualization techniques. In: IEEE. *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*. [S.l.], 2012. p. 343–348. ISSN 1524-4547.
- DUBE, R. R.; DIXIT, S. K. Process-oriented complete requirement engineering cycle for generic projects. In: *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. New York, NY, USA: ACM, 2010. (ICWET '10), p. 194–197. ISBN 978-1-60558-812-4. <<http://doi.acm.org/10.1145/1741906.1741948>>.
- EDUVIRGES, J. R. Classificações documentárias: semelhanças e diferenças entre cdd e cdu. UESPI, Piauí, 2011.
- FERREIRA, S. et al. Reducing the risk of requirements volatility: findings from an empirical survey. *Journal of Software: Evolution and Process*, Wiley Online Library, v. 23, n. 5, p. 375–393, 2011. <<https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.515>>.
- GHANBARI, H.; SIMILÄ, J.; MARKKULA, J. Utilizing online serious games to facilitate distributed requirements elicitation. *Journal of Systems and Software*, Elsevier, v. 109, p. 32–49, 2015. ISSN 0164-1212.
- GIL, A. C. *Métodos e técnicas de pesquisa social*. [S.l.]: 6. ed. Editora Atlas SA, 2008.
- GILL, P. et al. Methods of data collection in qualitative research: interviews and focus groups. *British dental journal*, Nature Publishing Group, v. 204, n. 6, p. 291, 2008.

- GOGUEN, J. A.; LINDE, C. Techniques for requirements elicitation. In: IEEE. *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*. [S.l.], 1993. p. 152–164.
- GUNDA, S. G. *Requirements engineering: elicitation techniques*. 2008.
- HANSEN, S.; BERENTE, N.; AVITAL, M. A first step toward innovative projects: An appreciative approach to requirements elicitation. Citeseer, 10 2014.
- ILYAS, F. et al. A survey on current requirement process practices in software companies and requirement process problems. In: IEEE. *Computational Science and Computational Intelligence (CSCI), 2016 International Conference on*. [S.l.], 2016. p. 1280–1285.
- INAYAT, I. et al. A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, Elsevier, v. 51, p. 915–929, 2015. ISSN 0747-5632.
- ISABIRYE, N.; FLOWERDAY, S. A model for eliciting user requirements specific to south african rural areas. In: ACM. *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*. New York, NY, USA: ACM, 2008. (SAICSIT '08), p. 124–130. ISBN 978-1-60558-286-3. <<http://doi.acm.org/10.1145/1456659.1456674>>.
- ISO/IEC/IEEE INTERNATIONAL STANDARD. Iso/iec/ieee international standard - systems and software engineering - life cycle processes - requirements engineering. *ISO/IEC/IEEE 29148:2011(E)*, p. 1–94, Dec 2011.
- JARKE, M.; POHL, K. Requirements engineering in 2001:(virtually) managing a changing reality. *Software Engineering Journal*, IET, v. 9, n. 6, p. 257–266, 1994.
- JWO, J.-S.; CHENG, Y. C. Pseudo software: A mediating instrument for modeling software requirements. *Journal of Systems and Software*, Elsevier, v. 83, n. 4, p. 599–608, 2010. ISSN 0164-1212.
- KASSAB, M.; NEILL, C.; LAPLANTE, P. State of practice in requirements engineering: contemporary data. *Innovations in Systems and Software Engineering*, Springer, v. 10, n. 4, p. 235–241, 2014.
- KHAN, K. et al. Requirement development life cycle: The industry practices. In: IEEE. *Software Engineering Research, Management and Applications (SERA), 2011 9th International Conference on*. [S.l.], 2011. p. 12–16.
- KHAN¹, S.; DULLOO, A. B.; VERMA, M. *Systematic review of requirement elicitation techniques*. [S.l.]: India, 2014.
- KITCHENHAM, B. Guidelines for performing systematic literature reviews in software engineering. ebse technical report. keele university & department of computer science university of durham. 2007.
- KITCHENHAM, B. What's up with software metrics?—a preliminary mapping study. *Journal of systems and software*, Elsevier, v. 83, n. 1, p. 37–51, 2010.

- KUCHMISTAYA, S. B. *Incorporation of Joint Application Design (JAD) in Systems Requirement Determination*. [S.l.]: University of Missouri, Information Systems Analysis Notes, November, 2001.
- LANGRIDGE, D. *Classificação: abordagem para estudantes de biblioteconomia*. [S.l.]: Interciência, 1977.
- LARAVEL. *Installation - Laravel - The PHP Framework For Web Artisans*. 2018. <<https://laravel.com/docs>>. Acessado em 12/10/2018.
- LAUESEN, S. Software requirements-styles and techniques. p. 344–345, 09 2017.
- LEITE, J. Engenharia de requisitos: Notas de aula. *Rio de Janeiro, PUC-Rio*, 1994.
- LEITE, J. C. S. P. *Viewpoint resolution in requirements elicitation*. Tese (Doutorado) — University of California, Irvine, 1988.
- LEITE, J. C. S. P. *Utilização de uma estratégia para identificação de fontes de informação na fase de elicitação*. Tese (Doutorado) — PUC-Rio, 2008.
- LI, Q.; CHEN, Y.-L. Entity-relationship diagram. In: *Modeling and Analysis of Enterprise and Information Systems*. [S.l.]: Springer, 2009. p. 125–139.
- MCLELLAN, S.; MUDDIMER, A.; PERES, S. C. The effect of experience on system usability scale ratings. *Journal of usability studies*, Usability Professionals' Association, v. 7, n. 2, p. 56–67, 2012.
- MILLS, J. Faceted classification and logical division in information retrieval. Graduate School of Library and Information Science. University of Illinois at Urbana-Champaign., 2004.
- MRAYAT, O. I. A.; NORWAWI, N.; BASIR, N. Requirements elicitation techniques: comparative study. *International Journal of Recent Development in Engineering and Technology*, v. 1, n. 3, p. 1–10, 2013.
- NATIONAL HEALTH AND MEDICAL RESEARCH COUNCIL. Book, Online. *How to review the evidence : systematic identification and review of the scientific literature : handbook series on preparing clinical practice guidelines*. National Health and Medical Research Council Canberra, 2000. xi, 112 p. : p. ISBN 1864960329. <<http://www.health.gov.au/hfs/nhmrc/publications/synopses/cp65syn.htm>>.
- NIJEM, Q. Software requirements elicitation tools for service oriented architecture: Comparative analysis. *International Journal of Computing Academic Research (IJCAR)* ISSN, Citeseer, p. 2305–9184, 2013.
- NUNES, L. et al. Da classificação das ciências à classificação da informação: uma análise do acesso ao conhecimento. Pontifícia Universidade Católica de Campinas, 2007.
- OLIVEIRA, A. L. C. et al. Breve ensaio sobre a evolução histórica da engenharia de requisitos: motivações para seu surgimento. *Universitas: Gestão e TI*, v. 2, n. 2, 2012.
- ORACLE CORP. *MySQL Documentation*. 2018. <<https://dev.mysql.com/doc/>>. Acessado em 12/10/2018.

- PETITCREW, M.; ROBERTS, H. *Systematic reviews in the social sciences: A practical guide*. [S.l.]: John Wiley & Sons, 2008.
- PIEDADE, M. A. R. *Introdução à teoria da classificação*. [S.l.]: Interciência, 1977.
- PRESSMAN, R. *Engenharia de Software uma abordagem profissional 7ª ed.* [S.l.]: McGraw Hill bookman AMGH Editora Ltda, 2011.
- PUTRA, L.; KANIGORO, B. et al. Design and implementation of web based home electrical appliance monitoring, diagnosing, and controlling system. *Procedia Computer Science*, Elsevier, v. 59, p. 34–44, 2015.
- RANABAHU, A. et al. A faceted classification based approach to search and rank web apis. In: IEEE. *Web Services, 2008. ICWS'08. IEEE International Conference on*. [S.l.], 2008. p. 177–184.
- REHMAN, T. ur; KHAN, M. N. A.; RIAZ, N. Analysis of requirement engineering processes, tools/techniques and methodologies. *International Journal of Information Technology and Computer Science (IJITCS)*, v. 5, n. 3, p. 40, 2013.
- SADIQ, M.; GHAFIR, S.; SHAHID, M. An approach for eliciting software requirements and its prioritization using analytic hierarchy process. In: IEEE. *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on*. [S.l.], 2009. p. 790–795.
- SAHA, S. K. et al. A systematic review on creativity techniques for requirements engineering. In: IEEE. *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*. [S.l.], 2012. p. 34–39.
- SENGUPTA, S.; BHATTACHARYA, S. Formalization of uml diagrams and their consistency verification: Az notation based approach. In: ACM. *Proceedings of the 1st India software engineering conference*. [S.l.], 2008. p. 151–152.
- SHARMA, S.; PANDEY, S. Revisiting requirements elicitation techniques. *International Journal of Computer Applications*, Foundation of Computer Science, v. 75, n. 12, 2013.
- SHARMA, S.; PANDEY, S. Requirements elicitation: Issues and challenges. In: IEEE. *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on*. [S.l.], 2014. p. 151–155.
- SILVA, E. L. d.; MENEZES, E. M. *Metodologia da pesquisa e elaboração de dissertação*. 3. ed. rev. atual, 2001.
- SILVA, M. B. d. et al. *A teoria da classificação facetada na modelagem de dados em banco de dados computacionais*. Universidade Federal da Paraíba, 2011.
- SOARES, H. A.; MOURA, R. S. A methodology to guide writing software requirements specification document. In: IEEE. *Computing Conference (CLEI), 2015 Latin American*. [S.l.], 2015. p. 1–11.
- SOLINGEN, R. v.; BERGHOUT, E. *The goal/question/metric method: a practical guide for quality improvement of software development*. McGraw-Hill Publishing Company, 1999.

SOMMERVILLE, I. *Engenharia de software. tradução: Selma shin shimuzu melnikoff, reginaldo arakaki, edilson de andrade barbosa*. [S.l.]: São Paulo: Pearson Addison Wesley, 2007.

SOMMERVILLE, I. *Engenharia de software. Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirama*-. [S.l.]: São Paulo: Pearson Prentice Hall, 2011.

SOUZA, K. S.; FANTINATO, M. Explorando a engenharia de requisitos orientada a serviços: Uma revisão sistemática da literatura. *Proceedings of the Simpósio Brasileiro de Sistemas de Informação*, 2013.

THE PHP GROUP. *PHP Hypertext Preprocessor*. 2018. <<http://php.net/>>. Acessado em 12/10/2018.

THE STANDISH GROUP. The standish group report-chaos. *The Standish Group*, 1995.

THE STANDISH GROUP. The standish group report-chaos. *The Standish Group*, 2015.

TOMAYKO, J. E. Engineering of unstable requirements using agile methods. In: *International Workshop on Time-Constrained Requirements Engineering (TCRE'02). Essen*. [S.l.: s.n.], 2002. v. 49.

TRISTÃO, A. M. D.; FACHIN, G. R. B.; ALARCON, O. E. Sistemas de classificação facetados e tesauros: instrumentos para organização do conhecimento. *Ciência da informação*, v. 33, n. 2, 2004.

TZITZIKAS, Y.; ANALYTI, A. Faceted taxonomy-based information management. In: IEEE. *Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on*. [S.l.], 2007. p. 207–211.

UMBER, A. et al. Requirements elicitation methods. In: TRANS TECH PUBL. *Advanced Materials Research*. [S.l.], 2012. v. 433, p. 6000–6006.

VIANNA, M. *Design thinking: inovação em negócios*. [S.l.]: Design Thinking, 2012.

VUEJS. *Introduction - Vue.js*. 2018. <<https://vuejs.org/v2/guide/index.html>>. Acessado em 12/10/2018.

WIEGERS, K.; BEATTY, J. *Software requirements*. [S.l.]: Pearson Education, 2013.

WOHLIN, C. et al. *Engineering and managing software requirements*. [S.l.]: Springer Science & Business Media, 2005.

YOUSEF, R.; ALMARABEH, T. An enhanced requirements elicitation framework based on business process models. *Scientific Research and Essays*, Academic Journals, v. 10, n. 7, p. 279–286, 2015.

YOUSUF, M.; ASGER, M. Comparison of various requirements elicitation techniques. *International Journal of Computer Applications*, Foundation of Computer Science, v. 116, n. 4, 2015.

ZHANG, Y.; WILDEMUTH, B. M. Unstructured interviews. *Applications of Social Research Methods to Questions in Information and Library Science*, ABC-CLIO, p. 000–060, 2016.

- ZHOU, X. et al. A map of threats to validity of systematic literature reviews in software engineering. In: IEEE. *Software Engineering Conference (APSEC), 2016 23rd Asia-Pacific*. [S.l.], 2016. p. 153–160.
- ZOWGHI, D.; COULIN, C. Requirements elicitation: A survey of techniques, approaches, and tools. In: *Engineering and managing software requirements*. [S.l.]: Springer, 2005. p. 19–46.

APÊNDICE A – STRINGS DE EXECUÇÃO DA BUSCA

Este apêndice se destina a apresentar as variações da string de busca base para a execução nas diversas bases científicas selecionadas. As bases estão listadas e descritas nas seções deste apêndice.

A.1 IEEEEXPLORE

- *Endereço da base:* ieeexplore.ieee.org
- *String de busca:* ((*"software requirements"*) AND (*"elicitation techniques"*OR *"acquisition techniques"*OR *"elicitation methods"*OR *"requirements elicitation"*) AND (*"evaluation"*OR *"comparison"*OR *"survey"*OR *"comparative study"*OR *"review"*OR *"mapping"*OR *"categories"*OR *"taxonomies"*OR *"classifications"*))
- *Observações:* Foi adicionado um parâmetro para o período conforme o critério de seleção.

A.2 ACM DIGITAL LIBRARY

- *Endereço da base:* dl.acm.org
- *String de busca:* ((*"software requirements"*) AND (*"elicitation techniques"*OR *"acquisition techniques"*OR *"elicitation methods"*OR *"requirements elicitation"*) AND (*"evaluation"*OR *"comparison"*OR *"survey"*OR *"comparative study"*OR *"review"*OR *"mapping"*OR *"categories"*OR *"taxonomies"*OR *"classifications"*))
- *Observações:* Foi adicionado um parâmetro para o período conforme o critério de seleção.

A.3 GOOGLE SCHOLAR

- *Endereço da base:* scholar.google.com.br
- *String de busca:* ((*"software requirements"*) AND (*"elicitation techniques"*OR *"acquisition techniques"*OR *"elicitation methods"*OR *"requirements elicitation"*) AND (*"evaluation"*OR *"comparison"*OR *"survey"*OR *"comparative study"*OR *"review"*OR *"mapping"*OR *"categories"*OR *"taxonomies"*OR *"classifications"*))
- *Observações:* Foi adicionado um parâmetro para o período conforme o critério de seleção. Esta base representa um agregador de resultados, desta forma foram selecionados os 100 primeiros resultados mais relevantes.

A.4 CITESEER LIBRARY

- *Endereço da base:* citeseer.ist.psu.edu
- *String de busca:* ((*"software requirements"*) AND (*"elicitation techniques"*OR *"acquisition techniques"*OR *"elicitation methods"*OR *"requirements elicitation"*) AND

("evaluation"OR "comparison"OR "survey"OR "comparative study"OR "review"OR "mapping"OR "categories"OR "taxonomies"OR "classifications"))

- *Observações:* No período em que as buscas foram executadas, a definição do período de publicação não estava funcionando adequadamente, por causa deste fator e pelo fato desta base ser um agregador de resultados, foram selecionados os 100 primeiros resultados mais relevantes que se adequam ao período definido nos critérios de seleção.

A.5 SCOPUS

- *Endereço da base:* www.scopus.com
- *String de busca:* (TITLE-ABS-KEY("software requirements") AND TITLE-ABS-KEY(("elicitation techniques"OR "acquisition techniques"OR "elicitation methods"OR "requirements elicitation"))AND TITLE-ABS-KEY(("evaluation"OR "comparison"OR "survey"OR "comparative study"OR "review"OR "mapping"OR "categories"OR "taxonomies"OR "classifications")))) AND (LIMIT-TO (PUBYEAR,2017) OR LIMIT-TO (PUBYEAR,2016) OR LIMIT-TO (PUBYEAR,2015) OR LIMIT-TO (PUBYEAR,2014) OR LIMIT-TO (PUBYEAR,2013) OR LIMIT-TO (PUBYEAR,2012) OR LIMIT-TO (PUBYEAR,2011) OR LIMIT-TO (PUBYEAR,2010) OR LIMIT-TO (PUBYEAR,2009) OR LIMIT-TO (PUBYEAR,2008)) AND (LIMIT-TO (SUBJAREA,"COMP"))

A.6 SPRINGER

- *Endereço da base:* www.springer.com
- *String de busca:* (("software requirements") AND ("elicitation techniques"OR "acquisition techniques"OR "elicitation methods"OR "requirements elicitation") AND ("evaluation"OR "comparison"OR "survey"OR "comparative study"OR "review"OR "mapping"OR "categories"OR "taxonomies"OR "classifications"))
- *Observações:* Foi adicionado um parâmetro para o período conforme o critério de seleção e foi definido um parâmetro para retornar somente publicações com acesso permitido.

A.7 SCIENCE DIRECT

- *Endereço da base:* www.sciencedirect.com
- *String de busca:* (("software requirements") AND ("elicitation techniques"OR "acquisition techniques"OR "elicitation methods"OR "requirements elicitation") AND ("evaluation"OR "comparison"OR "survey"OR "comparative study"OR "review"OR "mapping"OR "categories"OR "taxonomies"OR "classifications"))
- *Observações:* Foi definido na busca, um parâmetro relacionado ao período de publicação conforme os critérios de seleção.

A.8 ENGINEERING VILLAGE

- *Endereço da base:* www.engineeringvillage.com
- *String de busca:* (((((((("software requirements") AND ("elicitation techniques"OR "acquisition techniques"OR "elicitation methods"OR "requirements elicitation") AND ("evaluation"OR "comparison"OR "survey"OR "comparative study"OR "review"OR "mapping"OR "categories"OR "taxonomies"OR "classifications")))) WN ALL)) AND ((2017 OR 2016 OR 2015 OR 2014 OR 2013 OR 2012 OR 2011 OR 2010 OR 2009 OR 2008) WN YR)) AND (english WN LA)

APÊNDICE B – PUBLICAÇÕES SELECIONADAS

Este apêndice se destina a apresentar as publicações selecionadas na revisão sistemática que foram utilizadas na coleta das técnicas de elicitação de requisitos e suas respectivas características com o objetivo de responder as questões levantadas. As publicações estão compiladas na **Tabela 25**.

Tabela 25 – Publicações selecionadas na revisão sistemática

#	Referência	Título
P01	(ISABIRYE; FLOWERDAY, 2008)	A model for eliciting user requirements specific to South African rural areas
P02	(DUBE; DIXIT, 2010)	Process-oriented Complete Requirement Engineering Cycle for Generic Projects
P03	(HANSEN; BERENTE; AVITAL, 2014)	A First Step toward Innovative Projects: An Appreciative Approach to Requirements Elicitation
P04	(ABBASI et al., 2015)	Assessment of requirement elicitation tools and techniques by various parameters
P05	(YOUSUF; ASGER, 2015)	Comparison of various requirements elicitation techniques
P06	(UMBER et al., 2012)	Requirements Elicitation Methods
P07	(SHARMA; PANDEY, 2013)	Revisiting requirements elicitation techniques
P08	(INAYAT et al., 2015)	A systematic literature review on agile requirements engineering practices and challenges
P09	(SAHA et al., 2012)	A systematic review on creativity techniques for requirements engineering
P10	(YOUSEF; ALMARABEH, 2015)	An enhanced requirements elicitation framework based on business process models
P11	(REHMAN; KHAN; RIAZ, 2013)	Analysis of requirement engineering processes, tools/techniques and methodologies
P12	(BESROUR; RAHIM; DOMINIC, 2014)	Assessment and evaluation of requirements elicitation techniques using analysis determination requirements framework
P13	(DUARTE et al., 2012)	Collaborative requirements elicitation with visualization techniques
P14	(DIRGAHAYU; SETIANI; ZUKHRI, 2014)	Information requirements engineering for specific content management systems
P15	(FERREIRA et al., 2011)	Reducing the risk of requirements volatility: findings from an empirical survey
P16	(MRAYAT; NORWAWI; BASIR, 2013)	Requirements elicitation techniques: comparative study
P17	(SHARMA; PANDEY, 2014)	Requirements elicitation: Issues and challenges
P18	(ARIF; GAHYUR, 2009)	Requirements engineering processes, tools/technologies, & methodologies
P19	(DIESTE; JURISTO, 2011)	Systematic review and aggregation of empirical studies on elicitation techniques
P20	(KHAN ¹ ; DULLOO; VERMA, 2014)	Systematic review of requirement elicitation techniques
P21	(CARRIZO; DIESTE; JURISTO, 2014)	Systematizing requirements elicitation technique selection
P22	(DIESTE; JURISTO; SHULL, 2008)	Understanding the customer: what do we know about requirements elicitation?
P23	(ILYAS et al., 2016)	A Survey on Current Requirement Process Practices in Software Companies and Requirement Process Problems
P24	(KASSAB; NEILL; LAPLANTE, 2014)	State of practice in requirements engineering: contemporary data
P25	(SADIQ; GHAFIR; SHAHID, 2009)	An approach for eliciting software requirements and its prioritization using analytic hierarchy process
P26	(JWO; CHENG, 2010)	Pseudo software: A mediating instrument for modeling software requirements
P27	(GHANBARI; SIMILÄ; MARKKULA, 2015)	Utilizing online serious games to facilitate distributed requirements elicitation
P28	(ALEGRE; AUGUSTO; CLARK, 2016)	Engineering context-aware systems and applications: A survey
P29	(NIJEM, 2013)	Software Requirements Elicitation Tools for Service Oriented Architecture: Comparative Analysis
P30	(KHAN et al., 2011)	Requirement development life cycle: The industry practices

Fonte: Elaboração do autor, 2018

APÊNDICE C – CASOS DE USO

Este apêndice apresenta todos os casos de uso utilizados para desenvolvimento do guia conforme citado na **Seção 5.1.4**.

C.1 UC01 - EFETUAR LOGIN

Autentica um usuário com o perfil de administrador ao painel administrativo (**Figura 27**).

Pré-condição: O administrador deve ter um cadastro na plataforma.

Pós-condição: Administrador autenticado.

Figura 27 – Página de login

Painel administrativo
Guia de Técnicas de Elicitação de Requisitos

Por favor informe seus dados de acesso.

☐ Manter conectado

Acessar

[Esqueceu sua senha?](#)

Fonte: Elaboração do autor, 2018

Cenários

Efetua login (*principal*)

- 1.O administrador acessa o endereço do painel administrativo
- 2.O administrador informa os dados de login.
- 3.O sistema valida os dados de acesso.
- 4.O sistema redireciona o administrador para a página principal do painel administrativo (**Figura 28**).

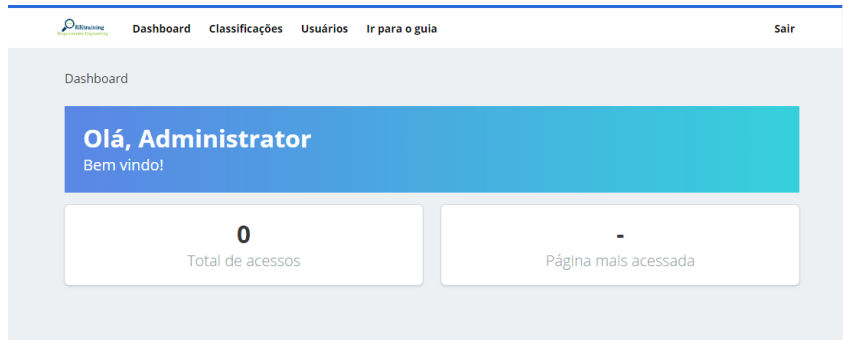
Manter conectado (*alternativo*)

No passo 1, se o administrador marcar a opção "Manter conectado", o sistema deve avaliar se existe uma sessão válida no navegador e autenticar o usuário automaticamente ao acessar o painel.

Dados de acesso inválidos (*exceção*)

No passo 2, se o login não existir ou estiver desativado ou se a senha não estiver correta o sistema apresenta a mensagem "Usuário e/ou Senha inválidos".

Figura 28 – Página principal



Fonte: Elaboração do autor, 2018

C.2 UC02 - RECUPERAR SENHA

Cria um link de alteração de senha para um usuário com perfil de administrador.

Pré-condição: O administrador deve ter um cadastro na plataforma.

Cenários

Gera o link para recuperação de senha (*principal*)

- 1.O administrador acessa o endereço do painel administrativo
- 2.O administrador acessa a opção "Esqueceu sua senha?".
- 3.O sistema apresenta um formulário solicitando o e-mail.
- 4.O administrador informa seu e-mail.
- 5.O sistema gera o link de recuperação de senha.
- 6.O sistema envia o link de recuperação para o e-mail do administrador.
- 7.O sistema apresenta a mensagem de sucesso.

E-mail inválido (*exceção*)

No passo 3, se o e-mail não existir ou estiver desativado o sistema apresenta a mensagem "E-mail inválido".

C.3 UC03 - ATUALIZAR SENHA ATRAVÉS DA RECUPERAÇÃO

Atualiza a senha do usuário através do endereço de recuperação.

Pré-condição: O usuário deve ter solicitado a recuperação de senha (UC03).

Pós-condição: Senha atualizada.

Cenários

Atualiza senha do administrador (*principal*)

- 1.O administrador acessa o endereço recebido por e-mail.

- 2.O sistema apresenta um formulário solicitando a nova senha e a confirmação da nova senha.
- 3.O administrador informa a nova senha de acesso e a confirmação da nova senha.
- 4.O administrador envia o formulário de alteração.
- 5.O sistema atualiza a senha do administrador.
- 6.O sistema autentica o administrador.
- 7.O sistema redireciona o administrador para a página principal do painel administrativo.

E-mail inválido (*exceção*)

No passo 3, se as senhas não forem iguais o sistema apresenta a mensagem "A confirmação de senha está incorreta".

C.4 UC04 - EFETUAR LOGOUT

Encerra a sessão de utilização do painel administrativo.

Pré-condição: O usuário deve estar autenticado no sistema.

Pós-condição: Sessão finalizada.

Cenários

Efetua o logout da aplicação (*principal*)

- 1.O administrador acessa a opção "Sair" no menu principal.
- 2.O sistema encerra sua sessão.
- 3.O sistema redireciona o administrador para a página de autenticação.

C.5 UC05 - GERENCIAR TÉCNICAS

Gerencia as técnicas apresentadas no guia conforme uma classificação.

Pré-condição: O administrador deve estar autenticado (UC01).

Cenários

Lista técnicas (Figura 29) (*principal*)

- 1.O administrador acessa a opção *Classificações* no menu principal.
- 2.O sistema apresenta as classificações disponíveis.
- 3.O administrador acessa a opção *Técnicas* da classificação desejada.
- 4.O sistema apresenta as técnicas cadastradas.

Figura 29 – Listagem de técnicas

<div>  Dashboard Classificações Usuários Ir para o guia </div> <div>Sair</div>			
Dashboard / Classificações / Técnicas de Elicitação de Requisitos / Técnicas			
Técnicas			✓ Nova Técnica
#	Título	Descrição	Ações
1	Entrevista	É conhecida por ser a mais comum das técnicas de elicitação de requisitos. As entrevistas geralmente envolvem discussões individuais de um representante da equipe com o stakeholder do novo sistema.	✎ ■
2	Prototipação	Prototipação pode ser considerada como a confecção de uma versão inicial do produto, preparada para coletar o feedback das partes interessadas e levantar alterações que devem ser incorporadas na próxima edição.	✎ ■
3	Brainstorming	É uma discussão informal onde cada participante pode expressar suas ideias livremente para que um novo sistema seja desenvolvido.	✎ ■
4	Card sorting	O especialista classifica um conjunto de cartões em grupos, cada um com o nome de alguma entidade de domínio escrita ou representada nele e efetua as associações com a participação dos usuários.	✎ ■
5	Entrevista em grupo	Nesta técnica um grupo de stakeholders é convidado para uma consulta coletiva onde são abordados assuntos do estado atual das práticas executadas e do que se espera ser executado no projeto.	✎ ■
1	2	3	« Anterior Próxima »

Fonte: Elaboração do autor, 2018

Cadastra técnica (Figura 30) (*alternativo*)

- 1.O administrador acessa a listagem de técnicas.
- 2.O sistema apresenta as técnicas cadastradas e ações disponíveis.
- 3.O administrador acessa a opção *Nova técnica*.
- 4.O sistema apresenta o formulário de cadastro.
- 5.O administrador informa os dados da técnica.
- 6.O administrador executa a opção cadastrar.
- 7.O sistema cadastra a técnica.
- 8.O sistema redireciona o administrador para a listagem de técnicas.

Figura 30 – Formulário de cadastro de técnicas

Fonte: Elaboração do autor, 2018

Dados obrigatórios não preenchidos no cadastro (*exceção*)

No passo 6 do cenário Cadastra técnica, se algum dos campos obrigatórios não for preenchidos, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Atualiza técnica (*alternativo*)

- 1.O administrador acessa a listagem de técnicas.
- 2.O sistema apresenta as técnicas cadastradas e ações disponíveis.
- 3.O administrador acessa a opção *Editar* de alguma das técnicas apresentadas.
- 4.O sistema apresenta o formulário com os dados da técnica acessada.
- 5.O administrador altera as informações necessárias.
- 6.O administrador executa a opção alterar.
- 7.O sistema altera a técnica.
- 8.O sistema redireciona o administrador para a listagem de técnicas.

Dados obrigatórios removidos na alteração (*exceção*)

No passo 6 do cenário Atualiza técnica, se alguma informação obrigatória for removida, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Classifica técnica (*alternativo*)

- 1.O administrador acessa a listagem de técnicas.
- 2.O sistema apresenta as técnicas cadastradas e ações disponíveis.
- 3.O administrador acessa a opção *Editar* de alguma das técnicas apresentadas.
- 4.O sistema apresenta o formulário com os dados da técnica acessada.
- 5.O administrador acessa a aba *Classificação*.
- 6.O administrador relaciona as técnicas com as facetas cadastradas.
- 7.O sistema grava a classificação da técnica.
- 8.O sistema redireciona o administrador para a listagem de técnicas.
- 9.O sistema apresenta uma mensagem de sucesso.

Altera situação de publicação (*alternativo*)

- 1.O administrador acessa a listagem de técnicas.
- 2.O sistema apresenta as técnicas cadastradas e ações disponíveis.
- 3.O administrador acessa a opção *Publicar* ou *Despublicar* de alguma das técnicas apresentadas.
- 4.O sistema altera a situação da técnica.
- 5.O sistema apresenta uma mensagem de sucesso.

C.6 UC06 - GERENCIAR FACETAS

Gerencia as facetas apresentadas no guia conforme uma classificação.

Pré-condição: O administrador deve estar autenticado (UC01).

Cenários

Lista facetas (Figura 31) (*principal*)

- 1.O administrador acessa a opção *Classificações* no menu principal.
- 2.O sistema apresenta as classificações disponíveis.
- 3.O administrador acessa a opção *Facetas* da classificação desejada.
- 4.O sistema apresenta as facetas cadastradas.

Cadastra faceta (Figura 32) (*alternativo*)

- 1.O administrador acessa a listagem de facetas.
- 2.O sistema apresenta as facetas cadastradas e as ações disponíveis

Figura 31 – Listagem de facetas

#	Título	Descrição	Ações
1	Categoria	Categorias utilizadas nas publicações selecionadas na revisão sistemática.	
2	Fonte principal	Fonte principal responsável por fornecer insumos ou conduzir a técnica. [3]	
3	Tipo de técnica	Identifica qual a forma de extração de requisitos. [1]	
4	Tipo de dado	Identifica o tipo do dado extraído [1]	
5	Comunicação	Identifica a direção da comunicação entre os envolvidos. [1]	

Fonte: Elaboração do autor, 2018

- 3.O administrador acessa a opção *Nova faceta*.
- 4.O sistema apresenta o formulário de cadastro.
- 5.O administrador informa os dados da faceta
- 6.O administrador executa a opção cadastrar.
- 7.O sistema cadastra a faceta.
- 8.O sistema redireciona o administrador para a listagem de facetas.

Figura 32 – Formulário de cadastro de facetas

Definição

Valores possíveis

Referências

Título

Tipo de técnica

Slug

tipo-de-tecnica

Descrição

Identifica qual a forma de extração de requisitos. [1]

checkboxbutton

Tipo

Selecione um tipo

Alterar

Cancelar

Fonte: Elaboração do autor, 2018

Dados obrigatórios não preenchidos no cadastro (*exceção*)

No passo 6 do cenário Cadastra faceta, se algum dos campos obrigatórios não for preenchidos, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Atualiza faceta (*alternativo*)

- 1.O administrador acessa a listagem de facetas.
- 2.O sistema apresenta as facetas cadastradas e as ações disponíveis
- 3.O administrador acessa a opção *Editar* de alguma das facetas apresentadas.
- 4.O sistema apresenta o formulário com os dados da faceta acessada pré-carregados.
- 5.O usuário altera as informações necessárias.
- 6.O administrador executa a opção alterar.
- 7.O sistema altera a faceta.
- 8.O sistema redireciona o administrador para a listagem de facetas.

Dados obrigatórios removidos na alteração (*exceção*)

No passo 6 do cenário Atualiza faceta, se alguma informação obrigatória for removida, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Exclui faceta (*alternativo*)

- 1.O administrador acessa a listagem de facetas.
- 2.O sistema apresenta as facetas cadastradas e as ações disponíveis
- 3.O administrador acessa a opção *Ecluir* de alguma das facetas apresentadas.
- 4.O sistema remove a faceta selecionada e suas associações com as técnicas.
- 5.O sistema redireciona o administrador para a listagem de facetas.
- 6.O sistema apresenta uma mensagem de sucesso.

C.7 UC07 - GERENCIAR CLASSIFICAÇÕES

Gerencia as classificações apresentadas no guia.

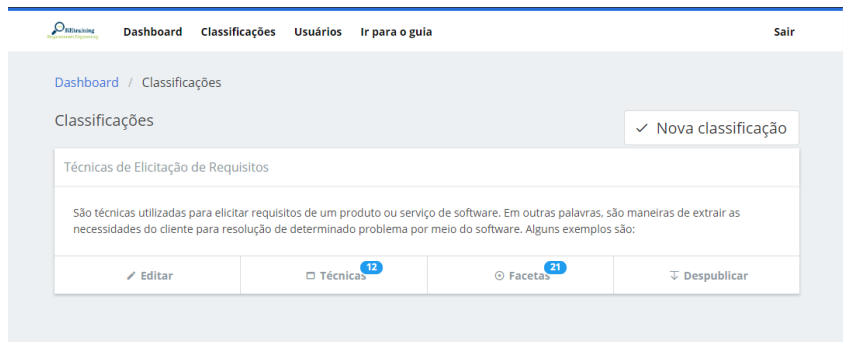
Pré-condição: O administrador deve estar autenticado (UC01).

Cenários

Lista classificações (Figura 33) (*principal*)

- 1.O administrador acessa a opção *Classificações* no menu principal.
- 2.O sistema apresenta as classificações cadastradas.

Figura 33 – Listagem de classificações



Fonte: Elaboração do autor, 2018

Cadastra classificação (Figura 34) (*alternativo*)

- 1.O administrador acessa a listagem de classificações.
- 2.O sistema apresenta as classificações cadastradas e as ações disponíveis.
- 3.O administrador acessa a opção *Nova classificação*.
- 4.O sistema apresenta o formulário de cadastro.
- 5.O administrador informa os dados da classificação.
- 6.O administrador executa a opção cadastrar.
- 7.O sistema cadastra a classificação.
- 8.O sistema redireciona o administrador para a listagem de classificações.

Figura 34 – Formulário de cadastro de classificações

Fonte: Elaboração do autor, 2018

Dados obrigatórios não preenchidos no cadastro (*exceção*)

No passo 6 do cenário Cadastra classificação, se algum dos campos obrigatórios não for preenchidos, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Atualiza classificação (*alternativo*)

- 1.O administrador acessa a listagem de classificações.
- 2.O sistema apresenta as classificações cadastradas e as ações disponíveis.
- 3.O administrador acessa a opção *Editar* de alguma das classificações apresentadas.
- 4.O sistema apresenta o formulário com os dados da classificação pré-carregados.
- 5.O administrador altera as informações necessárias.
- 6.O administrador executa a opção alterar.
- 7.O sistema altera a classificação.
- 8.O sistema redireciona o administrador para a listagem de classificações.

Dados obrigatórios removidos na alteração (*exceção*)

No passo 6 do cenário Atualiza classificação, se alguma informação obrigatória for removida, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Altera situação de publicação (*alternativo*)

- 1.O administrador acessa a listagem de classificações.
- 2.O sistema apresenta as classificações cadastradas e ações disponíveis.
- 3.O administrador acessa a opção *Publicar* ou *Despublicar* de alguma das classificações apresentadas.
- 4.O sistema altera a situação da classificação.
- 5.O sistema apresenta uma mensagem de sucesso.

C.8 UC08 - GERENCIAR USUÁRIOS DO PAINEL ADMINISTRATIVO

Gerencia os usuários que têm acesso ao painel administrativo.

Pré-condição: O administrador deve estar autenticado (UC01).

Cenários

Lista usuários (Figura 35) (*principal*)

- 1.O administrador acessa a opção *Usuários* no menu principal.
- 2.O sistema apresenta os usuários cadastrados.

Figura 35 – Listagem de usuários

#	Nome	E-mail	Ações
1	Administrator	admin@mailinator.com	

Dashboard / Usuários

Usuários ✓ Novo usuário

1

Fonte: Elaboração do autor, 2018

Cadastra usuário (Figura 36) (*alternativo*)

- 1.O administrador acessa a listagem de usuários.
- 2.O sistema apresenta os usuários cadastrados e ações disponíveis.
- 3.O administrador acessa a opção *Novo usuário*.
- 4.O sistema apresenta o formulário de cadastro.
- 5.O administrador informa os dados do usuário.
- 6.O administrador executa a opção cadastrar.
- 7.O sistema cadastra o usuário.
- 8.O sistema redireciona para a listagem de usuários.

Figura 36 – Formulário de cadastro de usuários

Dashboard / Usuários / Novo usuário

Nome

Informe o nome

E-mail

Informe o e-mail

Senha

Informe a senha

Cadastrar Cancelar

Fonte: Elaboração do autor, 2018

Dados obrigatórios não preenchidos no cadastro (*exceção*)

No passo 6 do cenário Cadastra usuário, se algum dos campos obrigatórios não for preenchidos, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Atualiza usuário (*alternativo*)

- 1.O administrador acessa a listagem de usuários.
- 2.O sistema apresenta os usuários cadastrados e ações disponíveis.
- 3.O administrador acessa a opção *Editar* de algum dos usuários apresentados.
- 4.O sistema apresenta o formulário com os dados do usuário pré-carregados.
- 5.O administrador altera as informações necessárias.
- 6.O administrador executa a opção alterar.
- 7.O sistema altera o usuário.
- 8.O sistema redireciona para a listagem de usuários.

Dados obrigatórios removidos na alteração (*exceção*)

No passo 6 do cenário Atualiza usuário, se alguma informação obrigatória for removida, o sistema deve apresentar o erro de não preenchimento indicando a informação esquecida.

Exclui usuário (*alternativo*)

- 1.O administrador acessa a listagem de usuários.
- 2.O sistema apresenta os usuários cadastrados e ações disponíveis.
- 3.O administrador acessa a opção *Excluir* de algum dos usuários apresentados.
- 4.O sistema remove o usuário.
- 5.O sistema redireciona para a listagem de usuários.
- 6.O sistema apresenta uma mensagem de sucesso.

C.9 UC09 - LISTAR TÉCNICAS

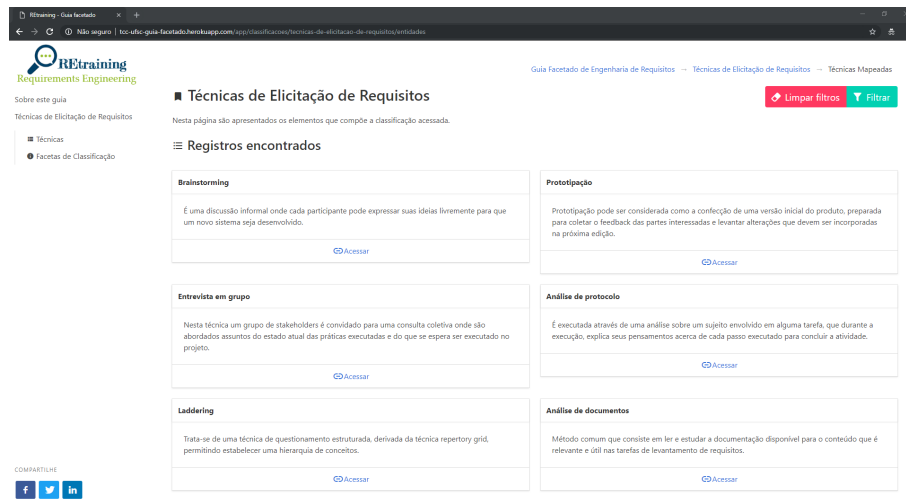
Lista as técnicas publicadas no guia.

Cenários

Visualiza técnicas publicadas (Figura 37) (*principal*)

- 1.O usuário acessa a opção *Técnicas* no menu principal.

Figura 37 – Listar técnicas através do Guia



Fonte: Elaboração do autor, 2018

2.O sistema apresenta as técnicas publicadas.

Apresenta mensagem caso não encontrar técnicas com base nos filtros (alternativo)

No passo 1, se o usuário executar a ação *Limpar filtros*, o sistema deve remover os filtros selecionados e apresentar as técnicas publicadas.

Visualiza técnicas publicadas com base nos filtros (Figura 38) (alternativo).

- 1.O usuário executa a opção *Filtrar* na listagem de técnicas.
- 2.O sistema apresenta os filtros disponíveis
- 3.O usuário seleciona os filtros.
- 4.O usuário executa a opção *Filtrar*.
- 5.O sistema busca as técnicas conforme os filtros selecionados.
- 6.O sistema apresenta as técnicas encontradas.

Apresenta mensagem caso não encontrar técnicas com base nos filtros (exceção)

No passo 4, dado que o usuário utilizou algum filtro, se o sistema não encontrar técnicas, deve apresentar a mensagem *Nenhum registro encontrado para o filtro informado*.

C.10 UC10 - VISUALIZAR TÉCNICAS

Visualizar as informações de uma técnica selecionada.

Cenários

Visualiza uma técnica selecionada (Figura 39) (principal)

Figura 38 – Página de filtros disponíveis

Seleção de filtros conforme as seguintes facetas

Geral

Categoria:

Fonte principal:

Elicitor

Treinamento na técnica de elicitação:

Experiência do elicitor:

Experiência com técnicas de elicitação:

Familiaridade com o domínio:

Domínio do problema

Tipo de dado:

Comunicação:

Tipo de informação a elicitar:

Nível de informação disponível:

Definição do problema:

Stakeholder

Pessoas por sessão:

Consenso entre os stakeholders:

Interesse do stakeholder:

Especialidade:

Articulação:

Disponibilidade de tempo:

Local/Acessibilidade:

Características gerais da técnica

Tipo de técnica:

Restrição de tempo do projeto:

Tempo de processo:

Fonte: Elaboração do autor, 2018

- 1.O usuário acessa a opção *Técnicas* no menu principal.
- 2.O sistema apresenta as técnicas disponíveis.
- 3.O usuário executa a opção *Acessar* de uma das técnicas apresentadas.
- 4.O sistema atualiza a contagem de acessos à técnica.
- 5.O sistema apresenta as informações cadastradas para a técnica acessada.

Figura 39 – Visualizar a página de uma técnica

Brainstorming

É uma discussão informal onde cada participante pode expressar suas ideias livremente para que um novo sistema seja desenvolvido. Os participantes devem ser de diferentes áreas e cada um possui um tempo para compartilhar suas ideias. Esta técnica tem foco na resolução de um problema particular. É comumente utilizada para definir o principal valor a ser entregue. Permite que os stakeholders pensem "fora da caixa" acerca do problema a ser resolvido, aumentando o potencial criativo das ideias. Não são permitidas críticas severas, pois podem causar uma tendência associada às respostas dadas. As ideias geradas são gravadas, as inapropriadas são descartadas e as apropriadas são priorizadas.

A aplicação da técnica é dividida em duas fases:

- Fase de geração, onde as ideias são coletadas sem nenhum julgamento ou critério.
- Fase de evolução, onde as ideias coletadas são discutidas.

Exemplos

Um exemplo de uma sessão de Brainstorming seria a seguinte:

- Situação:** O número de clientes que frequentam uma dada loja reduziu drasticamente nos meses recentes; o seu dono procura novas formas de atrair novos clientes e de aumentar a frequência de ida à loja dos clientes atuais

Prós

- Os custos são muito baixos e não são necessários muitos recursos.
- Todo participante é ativo no processo e não precisa ser altamente qualificado.
- É facilmente compreendida e de fácil execução.
- Ajuda na geração de novas ideias.
- Ajuda na resolução de conflitos.
- Todos participantes têm permissão de falar e compartilhar suas ideias igualmente.

Problema: Como atrair mais clientes?

A sessão de Brainstorming

- A sessão deve ser planejada. Defina o espaço e o horário.
- Os empregados na loja são convidados a participar. Todavia, o supervisor e o dono não tomarão parte da sessão.
- A sessão terá lugar longe da loja e num ambiente relaxado.
- O moderador deverá ser um bom comunicador e irá tentar quebrar o gelo e preparar o grupo com um jogo de associação de palavras.
- O moderador levanta a questão: Como podemos atrair mais clientes? É escreva a questão num quadro.
- O moderador explica as regras da sessão - 20 minutos, todos os participantes devem apresentar ideias e respeitar as dos colegas. **CRITICAR É PROIBIDO!**
- O moderador pede que sejam avançadas ideias.
- As ideias são produzidas em um mapa mental.
- Cada ideia é pontuada e avaliada pelos integrantes da sessão.

Contras

- Não é adequado para resolver problemas importantes.
- Se não for devidamente organizada pode consumir muito tempo.
- A quantidade de ideias não é proporcional a sua qualidade.
- Pode levar a repetição de ideias caso os participantes não estiverem prestando a devida atenção.
- Algumas pessoas podem ter medo de compartilhar suas ideias por causa da natureza extrovertida das discussões ou quando são criticadas.

Fonte: Elaboração do autor, 2018

C.11 UC11 - VISUALIZAR FACETAS

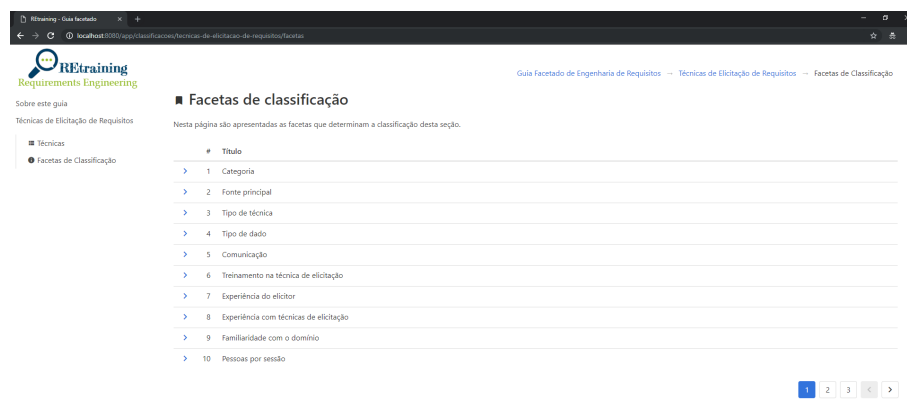
Visualizar as facetas cadastradas.

Cenários

Visualiza facetas cadastradas (Figura 40) (principal)

- 1.O usuário acessa a opção *Facetas de classificação* no menu principal.
- 2.O sistema apresenta as facetas cadastradas e suas respectivas informações.

Figura 40 – Visualizar facetas cadastradas



Fonte: Elaboração do autor, 2018

C.12 UC12 - COMPARTILHAR GUIA NAS REDES SOCIAIS

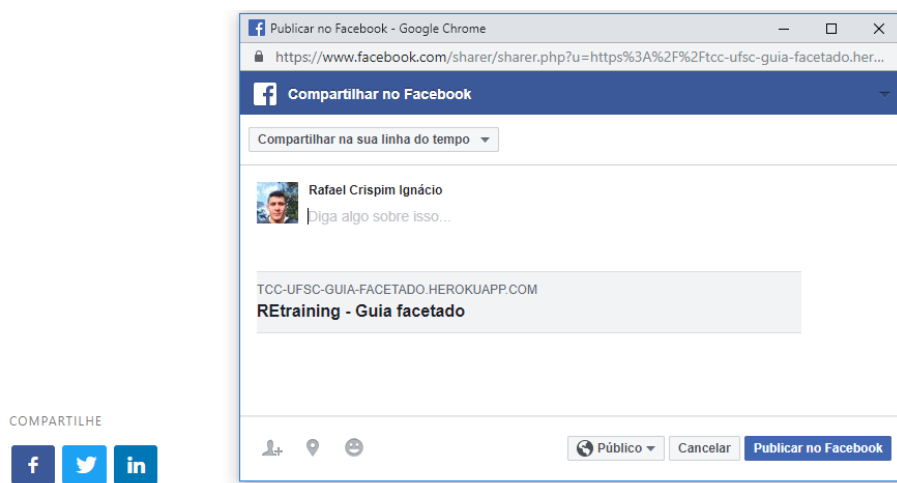
Compartilha página do Guia nas redes sociais.

Cenários

Compartilha o guia nas redes sociais (Figura 41) (principal)

- 1.O usuário acessa uma das opções de compartilhamento no menu principal.
- 2.O sistema apresenta um janela associado a rede selecionada.
- 3.O usuário efetua o compartilhamento.

Figura 41 – Compartilhamento com redes sociais



Fonte: Elaboração do autor, 2018

APÊNDICE D – ARTEFATOS UTILIZADOS NA AVALIAÇÃO

Este apêndice apresenta todos artefatos utilizados para a avaliação do trabalho executado.

D.1 TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Eu, *Nome do entrevistado*, concordo em participar da avaliação do Guia Facetado de Técnicas de Elicitação de Requisitos que tem como objetivo avaliar o conteúdo desenvolvido e os requisitos de usabilidade da plataforma. A participação nesse estudo consiste em responder algumas perguntas acerca do tema e realizar a utilização do sistema, expressando a satisfação em relação ao uso do guia.

Declaro que fui devidamente informado(a) pelo pesquisador Rafael Crispim Ignácio sobre os objetivos e procedimentos do estudo que serão utilizados e sobre a confidencialidade da pesquisa. Foi-me garantido que posso retirar o consentimento a qualquer momento, sem precisar prestar esclarecimentos. Declaro ainda que recebi uma cópia desse Termo de Consentimento.

Florianópolis, *dia* de *mês* de *ano*.

Assinatura do entrevistado

D.2 ROTEIRO DAS ENTREVISTAS

As entrevistas foram aplicadas presencialmente ou através de vídeo conferência com o auxílio de uma ferramenta de formulário online para coleta dos dados. Após a apresentação e a explicação dos objetivos da entrevista devem ser os passos em sequência:

- Coleta da identificação do entrevistado.
- Coleta das métricas 1.1.1, 1.1.2 e 1.1.3 do objetivo 1
- Apresentação e utilização da ferramenta
- Coleta das demais métricas do objetivo 1
- Coleta das métricas do objetivo 2

A **Tabela 26** apresenta todas questões utilizadas na entrevista na ordem em que foram feitas conforme definido no **Capítulo 6**.

Tabela 26 – Questões utilizadas na entrevista

#	Questão
Q1	Qual o seu nome?
Q2	Em qual empresa você trabalha?
Q3	Você possui formação superior completa?
Q4	Você possui um curso superior em andamento?
Q5	Qual o curso que você fez ou está fazendo?
Q6	Você já cursou disciplinas associadas à Engenharia de requisitos?
Q7	Você executa ou já executou atividades relacionadas a Engenharia de requisitos profissionalmente (licitação, análise, documentação, validação)?
Q8	Qual papel você frequentemente assume nos projetos?
Q9	Quanto tempo você teve ou tem com a atividade de levantamento/análise de requisitos?
Q10	Quais técnicas para elicitación (identificação) de requisitos você conhece?
Q11	Dado um estudo de caso (Tabela 16), quais técnicas você utilizaria para elicitación de requisitos?
Q12	Após a pesquisa no guia, quais técnicas você poderia utilizar para o estudo de caso avaliado anteriormente (Tabela 16)?
Q13	O conteúdo apresentado no guia é de fácil compreensão.
Q14	O conteúdo do guia é adequado (suficiente, claro e correto) em relação as técnicas acessadas.
Q15	Eu acho que gostaria de usar esse sistema com frequência.
Q16	Eu acho o sistema desnecessariamente complexo.
Q17	Eu achei o sistema fácil de usar.
Q18	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
Q19	Eu acho que as várias funções do sistema estão muito bem integradas.
Q20	Eu acho que o sistema apresenta muita inconsistência.
Q21	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
Q22	Eu achei o sistema atrapalhado de usar.
Q23	Eu me senti confiante ao usar o sistema.
Q24	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Fonte: Elaboração do autor, 2018

**APÊNDICE E – Guia Facetado de Técnicas de Elicitação de Requisitos -
Artigo**

Guia Facetado de Técnicas de Elicitação de Requisitos

Rafael Crispim Ignácio
Universidade Federal de Santa Catarina
Florianópolis
Santa Catarina
rafael.nacio@gmail.com

Fabiane Barreto Vavassori Benitti
Universidade Federal de Santa Catarina
Florianópolis
Santa Catarina
fabiane.benitti@ufsc.br

RESUMO

A engenharia de requisitos é uma das etapas mais importantes de um projeto, impactando direta ou indiretamente no sucesso do mesmo. Durante este processo, existem diversas técnicas que podem ser escolhidas para a elicitação e documentação dos requisitos. No cenário atual, não foram encontradas fontes, sejam elas artigos, pesquisas ou até guias, onde essas técnicas estão devidamente organizadas, facilitando o entendimento das mesmas na identificação suas características. A assertividade na escolha das técnicas pode influenciar na qualidade da fase de elicitação e por consequência no produto desenvolvido. Desta forma, este trabalho tem como objetivo principal levantar as técnicas de elicitação de requisitos existentes, organizando e categorizando de uma forma que auxilie na identificação de quais se adequam ao cenário de determinado projeto. Através do processo de revisão sistemática foram identificadas 96 técnicas de elicitação de requisitos. Para a classificação foram utilizadas 21 facetas distintas também com base no estudo executado. Por fim, no guia desenvolvido, são apresentadas 12 técnicas classificadas por meio de facetas.

Palavras-Chave

engenharia de software, elicitação de requisitos, classificação facetada, técnicas de elicitação.

ABSTRACT

Requirements engineering is one of the most important steps in a project, impacting directly or indirectly on its success. During this process, there are several techniques that can be chosen for the elicitation and documentation of requirements. In the current scenario, no sources were found, be they articles, surveys or even guides, where these techniques are properly organized, facilitating their understanding and get their characteristics. The assertiveness in the choice of techniques can influence the quality of the elicitation phase and consequently the developed product. In this way, this work has as main objective to raise the existing elicitation techniques, organizing and categorizing in a way that helps in identifying which ones fit the project scenario. Through the systematic review process, 96 elicitation techniques were identified. For the classification were used 21 distinct facets also based on the study performed. Finally, in the developed guide, 12 techniques classified by facets are presented.

Keywords

software engineering, requirements elicitation, faceted clas-

sification, elicitation techniques.

1. INTRODUÇÃO

A importância e a complexidade do desenvolvimento de software vem crescendo cada vez mais com o passar do tempo. Indivíduos, negócios e governos dependem, de forma crescente, de software para decisões estratégicas e táticas, assim como para controle e para operações cotidianas. Depreende-se, portanto, que um software deve apresentar qualidade elevada [15]. Assim, uma tarefa importante no desenvolvimento de software é a identificação e compreensão dos principais requisitos de negócios para assegurar que os produtos de software sejam totalmente compatíveis e evolutivos com o sistema [20]. A esta tarefa damos o nome de Engenharia de Requisitos [13], que possui o objetivo de criar e manter um documento de requisitos de sistema [17]. A Engenharia de requisitos pode ser dividida em quatro subprocessos [17], sendo um deles a Elicitação de Requisitos. Esta atividade é uma das partes mais importantes do processo, que pode levar ao sucesso ou ao fracasso do projeto [11]. Ela é considerada de grande importância pois em nenhum outro momento do projeto do sistema os interesses dos usuários e clientes cruzam-se mais do que no processo de elicitação de requisitos [19].

Dada a contextualização do tema e tendo em vista que o sucesso de um projeto de software será medido a partir da satisfação de seus usuários e partes interessadas no mesmo, faz-se necessário ter uma abordagem disciplinada para definir estas necessidades. Esta atividade pode ser feita de diversas formas e diversos fatores podem influenciar a mesma, notoriamente, questões relacionadas ao ambiente no qual a elicitação de requisitos é feita e a finalidade para qual o sistema se destina [14]. Como estas formas dependem de fatores externos e internos relacionados ao projeto, a decisão sobre quais técnicas aplicar pode se tornar um problema.

Este estudo pretende, através da definição de facetas, caracterizar as principais técnicas de elicitação de requisitos. Através da análise das técnicas existentes, é possível mapeá-las através de características comuns [5]. O conteúdo resultante possibilita uma decisão pautada nos aspectos do projeto em questão. Ao final, uma ferramenta de busca será desenvolvida tendo com base as técnicas levantadas e suas respectivas facetas. Seu principal objetivo é servir de interface para que um usuário possa, através dos fatores internos e externos do projeto, descobrir quais técnicas de elicitação de requisitos se enquadram melhor em sua situação. Desta forma, por meio do levantamento das técnicas de elicitação de requisitos, definição e aplicação de uma classificação fa-

cetada e do desenvolvimento de um guia, procura-se atingir o objetivo de desenvolver uma ferramenta de busca orientada a facetas para suporte a identificação de técnicas de elicitação de requisitos.

Este artigo está organizado da seguinte forma: Na seção 2 é apresentado o processo de revisão sistemática onde foram identificadas as técnicas de elicitação de requisitos. Na seção 3 compreende o desenvolvimento do guia facetado e o processo de classificação. Por fim, na seção 4 são expostos os resultados da realização da avaliação da ferramenta proposta.

2. TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

Esta seção tem o objetivo de identificar e analisar as técnicas de elicitação de requisitos existentes na bibliografia por meio de uma revisão sistemática. Uma revisão sistemática da literatura é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma determinada questão de pesquisa, área de tópico ou fenômeno de interesse [10].

2.1 Planejamento

Para realizar os objetivos foram formuladas duas questões utilizadas na revisão sistemática.

- Q1. Quais técnicas são utilizadas para elicitação de requisitos?
- Q2. Quais são as características das técnicas de Elicitação de Requisitos?

Na tabela 1 são apresentados os critérios utilizados na pesquisa.

Tabela 1: Critérios de pesquisa

A revisão deve considerar os artigos relacionados a área de requisitos de software com data de publicação de 01/01/2008 até 31/12/2017 (últimos 10 anos).

A partir da área de requisitos de software, o objetivo é estudar somente as técnicas de elicitação de requisitos e seus sinônimos.

Como critério de comparação foram levantados termos que levam a artigos que procuram agrupar técnicas de elicitação de requisitos.

Assim como no critério de comparação, no critério de resultados são esperados artigos que descrevem um conjunto de técnicas de elicitação de requisitos.

Por meio dos critérios definidos na tabela 1 foi desenvolvida a *string* de busca:

```
("software requirements") AND  
("elicitation techniques" OR "acquisition  
techniques" OR "elicitation methods" OR  
"requirements elicitation") AND  
("evaluation" OR "comparison" OR "survey"  
OR "comparative study" OR "review" OR  
"mapping" OR "categories" OR "taxonomies"  
OR "classifications")
```

A execução da busca deve considerar as seguintes bases [6]:

- IEEEExplore;
- ACM Digital Library;
- Google scholar;
- Citeseer library;
- ScienceDirect;

Com o intuito de complementar os resultados, outras bases, que também tem relação com o tema proposto, foram ser utilizadas:

- Springer;
- Scopus;
- Engineering Village;

Cada uma destas bases científicas possui uma forma específica de execução das buscas, desta forma, a *string* de busca foi adequada para cada caso respeitando os critérios estabelecidos no planejamento.

2.2 Execução

A seleção dos estudos é o processo no qual os estudos obtidos através das bases científicas são escolhidos para fazer parte do trabalho. Cabe a este processo avaliar os estudos em multicamadas, interpretando os critérios de seleção inicialmente no título e no resumo, se o trabalho em questão não for excluído, deve ser avaliado por completo [9]. Os artigos são selecionados com base em critérios pré-definidos que buscam identificar os estudos que fornecem evidências diretas sobre a questão de pesquisa [9].

2.2.1 Critérios de inclusão

Para o presente trabalho foram levantados os seguintes critérios de inclusão:

- Somente publicações com revisão por pares, como por exemplo conferências e journals;
- Deve contemplar 3 ou mais técnicas de elicitação, podendo ou não apresentar características das mesma;
- Deve ter sido publicado nos últimos 10 anos, de 01/01/2008 até 31/12/2017;

2.2.2 Critérios de exclusão

Da mesma forma foram elencados os critérios de exclusão, ou seja, características que definem que o estudo não deve ser selecionado:

- Publicações que não são revisadas por pares, como por exemplo, livros, teses, TCC's, etc;
- Publicações como resumos, pôsteres, tutoriais, serão excluídos;
- Estudos publicados em um idioma diferente do inglês;
- Artigos que não são acessíveis publicamente ou com autorização da universidade;

Além destes critérios, cabe ressaltar que se caso dois artigos apresentam o mesmo resultado, apenas o mais recente será considerado.

2.3 Resultados obtidos

Após a execução das buscas e coleta dos resultados, foram executados dois processos, sintetizados na **Figura 2**.

1. Análise dos títulos e resumos:

Ao analisar os títulos e resumos, foram removidos todos artigos que não tinham relação com o tema proposto e que consequentemente não seriam relevantes para responder as questões abordadas no presente trabalho. Nesta etapa, dos 453 artigos analisados, foram selecionados 222 artigos únicos, cerca de 50% do montante geral. Esta etapa utiliza condições mais superficiais, como a análise do assunto abordado pelo trabalho e seu objetivo. Este processo é rápido e diminui consideravelmente o número de trabalhos que precisam de uma análise mais detalhada;

2. Análise completa:

Nesta etapa são avaliados todos os trabalhos restantes de uma maneira mais detalhada, considerando todo o conteúdo dos estudos. Todos os critérios de seleção são considerados e definem se um artigo será selecionado ou não. Dos 222 artigos que passaram pela primeira etapa, foram selecionados 30 artigos únicos, cerca de 13% do montante geral.

A execução e seleção dos artigos gerou insumos para responder as questões levantadas no planejamento da revisão sistemática (seção 2.1).

2.3.1 Q1 - Quais técnicas são utilizadas para elicitação de requisitos?

Esta questão tem o objetivo de identificar as técnicas mais relevantes através dos trabalhos selecionados na revisão sistemática e obter informações inerentes às mesmas. Não existe a pretensão de atingir todas as técnicas existentes no meio acadêmico ou na indústria.

Através da análise dos estudos foram levantadas 96 técnicas de elicitação de requisitos. O número coletado demonstra que existem muitas técnicas catalogadas e que a maioria é pouco explorada, (59%) possui apenas 1 citação. Cerca de 21% das técnicas são encontradas em mais de 5 publicações. O estudo comprova também que a *Entrevista* é a técnica mais comum, encontrada em cerca de 83% das publicações selecionadas. Algumas técnicas podem possuir nomes diferentes, mas fazem referência ao mesmo método. Os casos identificados como similares através dos textos dos próprios autores foram unificados, caso contrário as técnicas foram mantidas com seus nomes de origem.

A **Figura 1** apresenta, através de uma nuvem de palavras, todas as técnicas encontradas, sendo que seus respectivos tamanhos na figura representam sua frequência perante as demais.

2.3.2 Q2 - Quais são as características das técnicas de Elicitação de Requisitos?

Esta questão tem o objetivo de identificar as características inerentes às técnicas de elicitação de requisitos a fim de agregar insumos para a criação de uma classificação facetada.

Também foram catalogadas 24 características através do estudo realizado. Estas informações foram extraídas de 4 entre 30 publicações selecionadas, representando cerca de



Figura 1: Nuvem de palavras de técnicas conforme a frequência na revisão sistemática

13%. A extração das características destes 4 artigos se deve ao fato dos autores especificarem as características diretamente, relacionando com as respectivas técnicas analisadas. Estas informações foram utilizadas para a definição da classificação das técnicas catalogadas no guia.

O número de técnicas coletadas e suas respectivas características representam a validade do planejamento da revisão sistemática. Estas informações fornecem insumos para que as técnicas possam ser devidamente classificadas perante suas características. Estes dados também mostram que a área de Elicitação de Requisitos está em constante de evolução, surgindo novas técnicas e derivações das técnicas existentes. Os métodos consolidados também são citados pela academia em vários dos artigos encontrados.

Os artigos selecionados representam, na maioria dos casos, estudos secundários. Estes casos foram selecionados através de uma revisão sistemática similar ao presente trabalho. Não foi encontrado na literatura, um artigo que tenha o mesmo objetivo deste trabalho, dificultando a comparação e validação do processo.

3. CLASSIFICAÇÃO FACETADA

Uma classificação facetada é conhecida como um esquema analítico sintético porque envolve dois processos distintos: a análise do assunto em facetas e a síntese dos elementos que constituem o mesmo, sendo, portanto, aplicável a qualquer área do conhecimento [18].

Neste trabalho foi utilizada uma metodologia que consiste na execução de sete passos [8]:

1. Coleta de Domínios: Coletar uma amostra representativa das entidades
2. Listagem de Entidades: Listar as entidades organizando os conceitos
3. Criação de Facetas: Examinar os termos resultantes, examinando e reduzindo em facetas mutuamente exclusivas e conjuntamente exaustivas
4. Arranjo de Facetas: Esta etapa contém duas tarefas. Primeiro deve ser feita uma ordem de teste de todos termos sob as facetas. Por fim, após testar e validar o rascunho das facetas, é necessário fazer um arranjo final

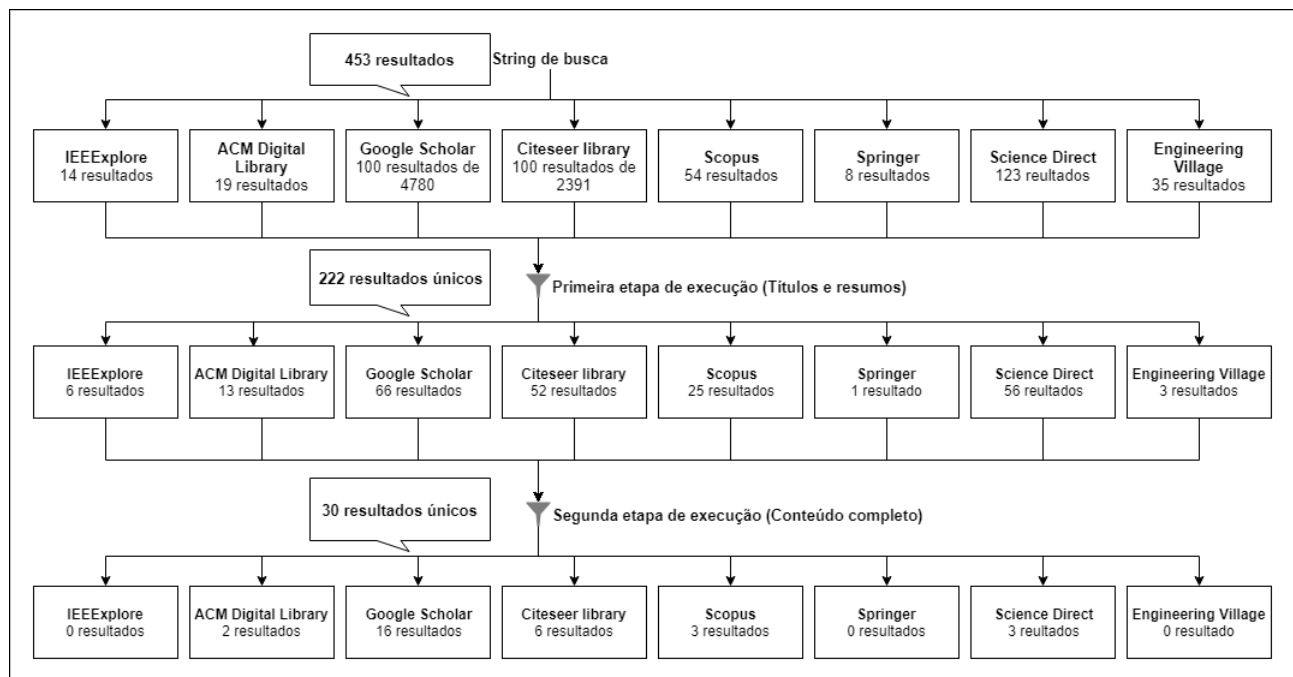


Figura 2: Detalhamento da execução da busca nas bases científicas.

5. Ordem de Citação: Esta fase faz parte do plano notacional. Diz respeito a qual ordem de citação padrão que será a maneira padrão como as coisas serão ordenadas
6. Classificação: Esta é a fase de execução onde o domínio será classificado conforme as facetas definidas
7. Revisão, teste e manutenção: Se for detectado algum problema na sexta etapa, deve-se voltar quantas etapas forem necessárias para ajustar a classificação

O domínio foi representado por todas técnicas identificadas no estudo (96). Foram utilizadas as características também coletadas através do estudo realizado para a derivação das facetas. Ao final, foram criadas 21 facetas por meio do qual é possível classificar as técnicas de elicitação de requisitos (tabela 2).

Apesar de ser um processo tecnicamente simples, a criação de uma classificação facetada se mostra complexa em alguns pontos. Existe uma dependência muito grande dos insumos utilizados na categorização do domínio, além de ser necessário um certo conhecimento na área e nos objetivos da classificação para saber como tratar a informação coletada. Outro fator muito importante está associado ao processo de teste e manutenção da estrutura da classificação facetada, que tende a refinar todo o trabalho desenvolvido impactando nos resultados alcançados através dos filtros.

4. DESENVOLVIMENTO DO GUIA

Esta seção possui o objetivo de especificar todas as decisões tomadas no desenvolvimento do guia. Os requisitos foram elicítados segundo os aspectos funcionais, não funcionais e de negócio. Para a elicitação de requisitos do presente trabalho, foram utilizadas as técnicas:

- Entrevistas: Para coletar os requisitos do projeto com a orientadora.

Categoria
Fonte principal
Treinamento na técnica de elicitação
Experiência do elicitor
Experiência com técnicas de elicitação
Familiaridade com o domínio
Tipo de informação a elicitar
Nível de informação disponível
Definição do problema
Tipo de dado
Comunicação
Pessoas por sessão
Consenso entre os stakeholders
Interesse do stakeholder
Especialidade
Articulação
Disponibilidade de tempo
Local/Acessibilidade
Tipo de técnica
Restrição de tempo do projeto
Tempo de processo

Tabela 2: Definição das facetas utilizadas na classificação

- Prototipação: Para validar as funcionalidades da aplicação e prever a organização das informações apresentadas.
- Análise de documentos: Para a coleta de informações utilizadas no Guia através da revisão sistemática apresentada na **Seção 2**.

Além das questões associadas à apresentação do guia na web, é importante ressaltar que um fator determinante para

a identificação dos requisitos não funcionais foi o conhecimento prévio em algumas das tecnologias utilizadas, como por exemplo a linguagem de programação utilizada na camada backend, este fator tem o objetivo de viabilizar o desenvolvimento do projeto em tempo hábil.

A aplicação está separada em duas camadas (figura 3). A primeira é apenas de visualização, onde é apresentado o guia. Esta camada foi desenvolvida utilizando HTML, CSS e Javascript por meio de um framework chamado VueJS [2]. A segunda camada consiste na estrutura do servidor da aplicação para responder as requisições efetuadas pelos usuários. Neste caso, além das linguagens de programação utilizadas na camada de visualização, foi utilizado o PHP [4] como linguagem principal com o framework Laravel [1] e o banco de dados MySQL [3] para persistência dos dados.

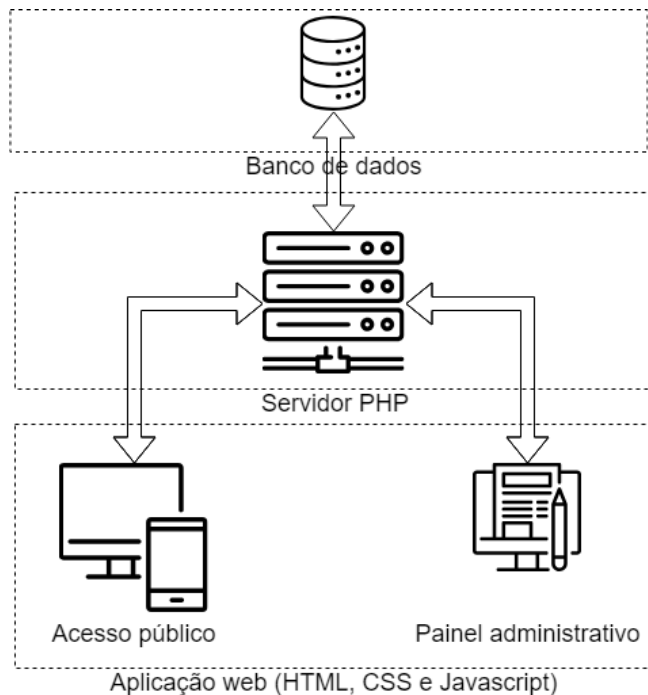


Figura 3: Estrutura do projeto desenvolvido.

Em relação ao servidor, foi utilizado a linguagem de programação PHP, através de um framework chamado Laravel, gratuito e de código aberto que utiliza MVC (Modelo, Visão e Controle) como padrão de projeto para desenvolvimento de aplicações web [16]. Até o momento do desenvolvimento da aplicação, a versão estável deste framework (5.6). Foram utilizadas todas as práticas presentes na documentação oficial para facilitar a configuração, manutenção e compreensão da ferramenta. Seus principais objetivos neste projeto são:

- Fornecer uma rota de entrada para o guia.
- Fornecer uma API de consumo para a aplicação cliente.
- Fornecer um painel administrativo para gestão do conteúdo apresentado.

No lado do cliente, além das tecnologias mais conhecidas para desenvolvimento de sistemas web, como HTML, CSS e Javascript, foi utilizado um framework chamado VueJS.

VueJS é um framework progressivo para desenvolvimento de interfaces para usuário, desenvolvido em javascript, gratuito e de código fonte aberto. Recebe o título de progressivo pois tem o foco na facilidade de integrar com outras bibliotecas ou projetos existentes [2]. Seu objetivo é fornecer uma interface de acesso ao guia, por onde os usuários consultam as informações disponibilizadas. Sua configuração também está detalhada no repositório do código do projeto. A figura 4 representa a versão disponibilizada ao público, acessível através do endereço <http://retraining.inf.ufsc.br/guia/app>.



Figura 4: Guia facetado de elicitação de requisitos.

5. AVALIAÇÃO

Com o intuito de avaliar as informações coletadas e o guia desenvolvido, agregando ainda mais valor para o presente trabalho, faz-se necessária a utilização de um método estabelecido para guiar a avaliação a ser realizada. A opção adotada neste caso foi a abordagem GQM (Goal-Question-Metric), um mecanismo para planejamento e definição dos objetivos a serem medidos e avaliados [7].

O modelo resultante possui 3 níveis [7]:

- Goal: Definição de uma meta para determinado objeto a ser avaliado no contexto do projeto.
- Question: Uma série de perguntas usadas para caracterizar uma meta específica de um objetivo que será realizado.
- Metric: Uma série de indicadores associados a cada pergunta realizada.

5.1 Definição

O primeiro objetivo diz respeito a qualidade e organização do conteúdo produzido. Desta forma, a fim de analisar o guia de técnicas de elicitação de requisitos, com o propósito de avaliar a facilidade de entendimento, com respeito a apresentação de novas técnicas e compreensão das mesmas, do ponto de vista de profissionais e acadêmicos, no contexto da elicitação de requisitos foram levantadas as seguintes questões e respectivas métricas (tabelas 3 e 4):

O segundo objetivo diz respeito ao Guia, mais especificamente quanto a usabilidade da ferramenta. Neste caso procura-se analisar o guia de técnicas de elicitação de requisitos, com o propósito de avaliar a facilidade de utilização da ferramenta, com respeito a navegação e disposição das

Tabela 3: 1.1 Questão: Quão efetivo é o guia em sugerir novas técnicas

M1.1.1) Quais técnicas para elicitación (identificação) de requisitos você conhece?
M1.1.2) Dado um caso de exemplo (Tabela A), quais técnicas você utilizaria para a elicitación de requisitos?
M1.1.3) Após a pesquisa no guia, quais técnicas você poderia utilizar para o caso de exemplo avaliado anteriormente (Tabela A)?

Tabela 4: 1.2 Questão: Qual o grau de compreensão das técnicas apresentadas pelo Guia de elicitación de requisitos

M1.1.1) O conteúdo apresentado no guia é de fácil compreensão.
M1.1.2) O conteúdo do guia é adequado (suficiente, claro e correto) em relação as técnicas acessadas.

informações, do ponto de vista de usuário, no contexto da aplicação. Para responder este objetivo foi utilizada a escala SUS que resulta em um indicador que corresponde ao nível de usabilidade da ferramenta avaliada [12]. Assim, foram aplicadas as questões apresentadas na tabela 5 para análise posterior.

Tabela 5: 2.1 Questão: Qual a pontuação na escala SUS obtida pela ferramenta

M2.1.1) Eu acho que gostaria de usar esse sistema com frequência.
M2.1.2) Eu acho o sistema desnecessariamente complexo.
M2.1.3) Eu achei o sistema fácil de usar.
M2.1.4) Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
M2.1.5) Eu acho que as várias funções do sistema estão muito bem integradas.
M2.1.6) Eu acho que o sistema apresenta muita inconsistência.
M2.1.7) Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
M2.1.8) Eu achei o sistema atrapalhado de usar.
M2.1.9) Eu me senti confiante ao usar o sistema.
M2.1.10) Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

5.2 Resultados obtidos

Foram selecionadas pessoas próximas ao autor, do meio acadêmico e do meio profissional, com ou sem experiência na atividade de elicitación de requisitos e que estão inseridas no meio de tecnologia da informação. Ao todo foram entrevistadas 16 pessoas, 12 destas apresentando mais de 1 ano de experiência na área de elicitación de requisitos.

Para responder sobre a efetividade do guia em sugerir novas técnicas (**Questão 1.1**) foram feitas perguntas acerca do tema, antes e depois da apresentação do guia. Primeiramente foram questionadas quais técnicas conhecidas pelo entrevistado. As respostas coletadas não apresentaram um número alto de técnicas e em alguns casos, os termos citados não eram adequados à etapa de Elicitación de Requisitos.

Em um segundo momento, a partir da apresentação de um caso de exemplo, foi perguntado aos entrevistados quais técnicas mais adequadas para a elicitación de requisitos, considerando as condições associadas ao problema em questão. A interpretação das características apresentadas também era parte do processo. Na maioria dos casos, as respostas não foram adequadas ao problema apresentado. Apenas 1 participante citou uma técnica esperada como uma opção válida (figura 5).

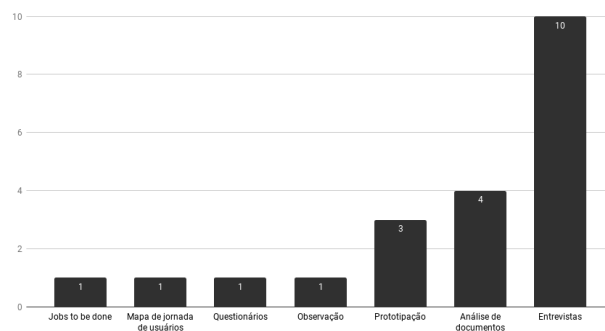


Figura 5: Técnicas citadas antes da apresentação do guia.

Após a apresentação do guia, o caso de exemplo foi reapresentado e novamente foi questionado quais técnicas poderiam ser utilizadas. Alguns entrevistados ainda apresentaram técnicas não muito adequadas, mas desta vez a maioria apresentou as respostas esperadas para o problema (figura 6).

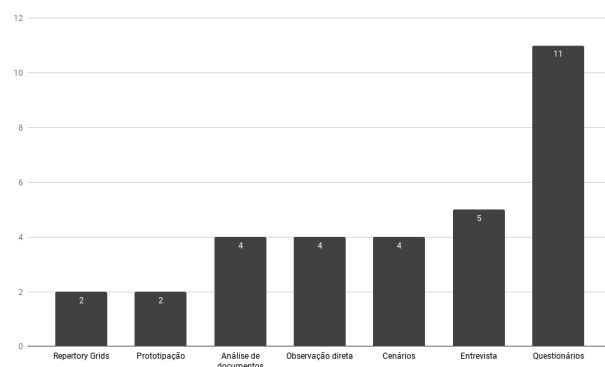


Figura 6: Técnicas citadas depois da apresentação do guia.

Cabe destacar também que, ao comparar as figuras 5 e 6, é possível perceber que cerca de 69% dos participantes mudaram de ideia após a utilização do guia, indicando uma certa influência do guia sobre suas respostas.

Na maioria dos casos (69%), os entrevistados informaram uma ou mais técnicas dentre as possíveis opções para o caso de exemplo apresentado. A quantidade de respostas adequadas aos resultados esperados, assim como uma comparação entre o momento anterior e posterior da apresentação do guia podem ser vistos na figuras 7 e 8.

A diferença entre os dois cenários é muito clara, pode-se

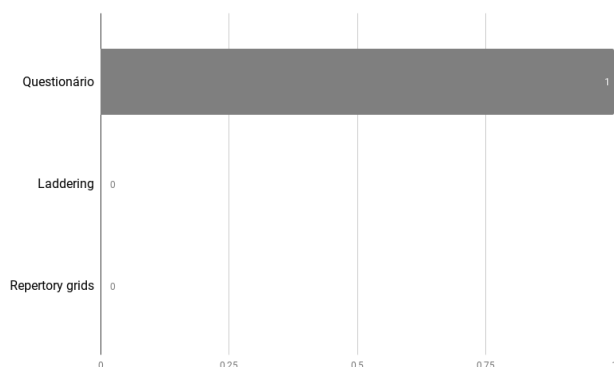


Figura 7: Total de resultados esperados antes da apresentação do guia.

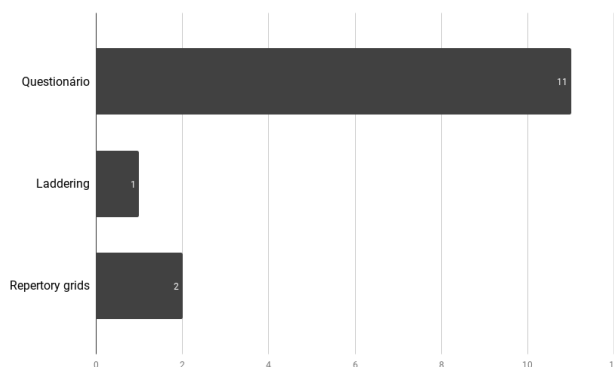


Figura 8: Total de resultados esperados após da apresentação do guia.

perceber que a ferramenta de fato auxiliou os participantes a chegarem em uma das respostas esperadas para o caso de exemplo. Por fim, é possível perceber também que metade dos participantes citaram em suas respostas, técnicas que ainda não estão presentes no guia. Na **Figura 9** são apresentadas as técnicas citadas pelos entrevistados que não constam no guia. *Análise de documentos* se destaca como a técnica que foi citada mais vezes e não foi classificada.

Complementando a avaliação do conteúdo apresentado, foram efetuadas perguntas acerca da qualidade e adequação do conteúdo produzido e apresentado no guia (**Questão 1.2**). Foram feitas perguntas utilizando uma escala Likert, onde os entrevistados poderiam optar entre, discordo totalmente, discordo parcialmente, concordo parcialmente e concordo totalmente para as afirmações associadas à compreensão do conteúdo e adequação do conteúdo em relação às técnicas acessadas. Para a primeira métrica desta questão, foi levantado que 88% dos participantes concordaram totalmente que a ferramenta possui um conteúdo de fácil compreensão, os demais entrevistados apenas concordaram parcialmente com essa afirmação. A proporção de entrevistados que concordaram totalmente com a afirmação de que o conteúdo é adequado para as técnicas acessadas se manteve em 88%, sendo que 6% concordaram parcialmente e outros 6% discordaram parcialmente.

Antes de entrar na análise das estatísticas levantadas, é

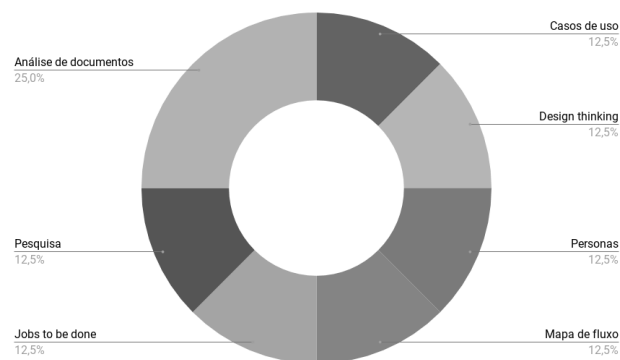


Figura 9: Técnicas citadas que não estão presentes no guia.

interessante destacar que, na maioria das vezes os entrevistados não tinham certeza ou não sabiam a nomenclatura das técnicas. Para estes casos os participantes foram incentivados a informar as técnicas da maneira que tinham conhecimento. Algumas respostas associadas às técnicas eram inadequadas ao tema, ou seja, não se tratavam de técnicas de elicitação de requisitos.

Como resultado do primeiro objetivo da avaliação, consideramos que o guia auxiliou os entrevistados apresentando técnicas que eles não conheciam ou não lembravam. Com o acesso à plataforma foi possível guiar a escolha de quais técnicas mais adequadas para utilização em determinado cenário. Os filtros também foram essenciais para encontrar as técnicas com base nas características do caso de exemplo apresentado, justificando a utilização da classificação facetada neste trabalho. O conteúdo produzido foi avaliado positivamente pela maioria dos participantes, tanto para as afirmações relacionadas a compreensão do conteúdo quanto para as afirmações associadas à adequação dos conteúdos de cada uma das técnicas apresentadas.

Quanto ao segundo objetivo de avaliação (tabela 5), foram analisados os aspectos de usabilidade associados à ferramenta desenvolvida. Para responder a pergunta relacionada a usabilidade, optou-se em utilizar a escala SUS, conforme apresentado na seção 5.1. O indicador gerado é calculado conforme cada uma das respostas dos participantes, de onde é extraída uma média. Para o guia apresentado, a escala SUS resultante foi de 90. Este valor indica que a plataforma possui um nível de usabilidade aceitável dentro da escala utilizada.

6. CONSIDERAÇÕES FINAIS

A pesquisa demonstrou que existem inúmeras técnicas de elicitação de requisitos, foram encontradas cerca de 96 técnicas conforme os critérios aplicados na revisão sistemática. Foram encontradas desde técnicas bem estabelecidas no mercado até técnicas que estão em fase de experimentação. Estes fatores validam as informações citadas por outros autores e citadas no início deste trabalho (**Seção 1**) referente a existência de um número considerável de técnicas e na viabilização de um guia condensando suas informações.

A utilização da classificação facetada neste trabalho demonstrou o quão simples e útil pode ser este método. A seleção das características a serem tratadas como facetas se

mostrou como a parte mais importante de todas. Uma falha nesta etapa pode acarretar em classificações incoerentes ou que não apresentem um resultado expressivo quanto ao agrupamento de seus elementos. Foram classificadas 12 técnicas de elicitação de requisitos, um número relativamente baixo se comparado às 96 técnicas encontradas no estudo, mas que já demonstraram resultados interessantes para as buscas efetuadas.

A avaliação da ferramenta apresentou resultados positivos tanto para os conteúdos produzidos quanto para a ferramenta desenvolvida, demonstrando que as decisões tomadas foram assertivas no decorrer da execução. Uma parte significativa dos entrevistados (50%) acabaram citando algumas técnicas que não estavam presentes no guia. Com isso, entende-se que é interessante que seu conteúdo seja atualizado com a inclusão de outras técnicas de elicitação de requisitos, com o objetivo de manter uma base mais completa possível. Além das estatísticas apresentadas na **Seção 5**, é importante mencionar que grande parte dos entrevistados apresentaram interesse que a ferramenta fosse divulgada quando publicada oficialmente.

Por fim, é importante destacar que, como acadêmico, este trabalho proporcionou grande aprendizado durante sua execução. Além do conhecimento adquirido sobre o tema proposto, ou seja, técnicas de elicitação de requisitos, classificação facetada e desenvolvimento do guia, o trabalho proporcionou um aprendizado acerca dos processos de pesquisa científica e seus derivados.

6.1 Trabalhos futuros

A partir dos resultados alcançados, ficam como sugestões de trabalhos a serem desenvolvidos:

- Coleta e classificação das demais técnicas encontradas a fim de completar o guia e fornecer uma base centralizada de técnicas conhecidas, enriquecendo ainda mais a plataforma e fornecendo insumos para os filtros realizados.
- Classificação de técnicas utilizadas em outras etapas da engenharia de requisitos, uma vez que para cada etapa deste processo existe uma variedade de técnicas que podem ser utilizadas.
- Revisão e avaliação de facetas para a composição de uma classificação mais completa e robusta, com características mais alinhadas com a necessidade dos usuários que procuram combinações para o seu cenário.
- Mapeamento das relações entre as técnicas de elicitação e apresentação no guia, uma vez que as técnicas podem ser utilizadas com objetivos diferentes, uma relação poderia ajudar a achar uma combinação ou o descarte da utilização de métodos similares.

7. REFERÊNCIAS

- [1] Installation - laravel - the php framework for web artisans, 2018.
- [2] Introduction - vue.js, 2018.
- [3] Mysql documentation, 2018.
- [4] Php hypertext preprocessor, 2018.
- [5] A. Belgamo, L. Eduardo, and L. E. Martins. Estudo comparativo sobre as técnicas de elicitação de requisitos do software. 01 2008.
- [6] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4):571–583, 2007.
- [7] V. R. B.-G. Caldiera and H. D. Rombach. Goal question metric paradigm. *Encyclopedia of software engineering*, 1:528–532, 1994.
- [8] W. Denton. How to make a faceted classification and put it on the web. See <http://www.miskatonic.org/library/facet-web-howto.html>, 2003.
- [9] B. Kitchenham. Guidelines for performing systematic literature reviews in software engineering. ebse technical report. keele university & department of computer science university of durham. 2007.
- [10] B. Kitchenham. What's up with software metrics?—a preliminary mapping study. *Journal of systems and software*, 83(1):37–51, 2010.
- [11] S. B. Kuchmistaya. Incorporation of joint application design (jad) in systems requirement determination, 2001.
- [12] S. McLellan, A. Muddimer, and S. C. Peres. The effect of experience on system usability scale ratings. *Journal of usability studies*, 7(2):56–67, 2012.
- [13] A. L. C. Oliveira, L. V. de Castro Teixeira, L. R. Vieira, and M. R. dos Santos. Breve ensaio sobre a evolução histórica da engenharia de requisitos: motivações para seu surgimento. *Universitas: Gestão e TI*, 2(2), 2012.
- [14] J. C. S. Prado Leite. *Utilização de uma estratégia para identificação de fontes de informação na fase de elicitação*. PhD thesis, PUC-Rio, 2008.
- [15] R. Pressman. Engenharia de software uma abordagem profissional 7^a ed, 2011.
- [16] L. Putra, B. Kanigoro, et al. Design and implementation of web based home electrical appliance monitoring, diagnosing, and controlling system. *Procedia Computer Science*, 59:34–44, 2015.
- [17] I. Sommerville. Engenharia de software. tradução: Selma shin shimuzu melnikoff, reginaldo arakaki, edilson de andrade barbosa, 2007.
- [18] A. M. D. Tristão, G. R. B. Fachin, and O. E. Alarcon. Sistemas de classificação facetados e tesauros: instrumentos para organização do conhecimento. *Ciência da informação*, 33(2), 2004.
- [19] K. Wiegers and J. Beatty. *Software requirements*. Pearson Education, 2013.
- [20] C. Wohlin et al. *Engineering and managing software requirements*. Springer Science & Business Media, 2005.

APÊNDICE

A. CASO DE EXEMPLO UTILIZADO NA AVALIAÇÃO

Este apêndice apresenta o caso de exemplo utilizado nas entrevistas da avaliação.

Tabela 6: Caso de exemplo para utilização na coleta de dados

Caso de exemplo:

Supondo que você trabalhe como analista em uma empresa bem estabelecida no mercado que desenvolve software para o ramo de supermercados*. A empresa acaba de aprovar um novo projeto de software para lojas de pequeno porte. As regras de negócio são bem similares ao que a empresa já conhece, facilitando no entendimento do problema**. Devido a alta temporada, os stakeholders acabam passando muito tempo em viagens negociando com fornecedores e não conseguem se reunir para definir os objetivos do projeto***.

** Contextualização da situação.

*** Condição que indica de que o analista e a empresa já possuem conhecimento prévio quanto ao problema abordado.

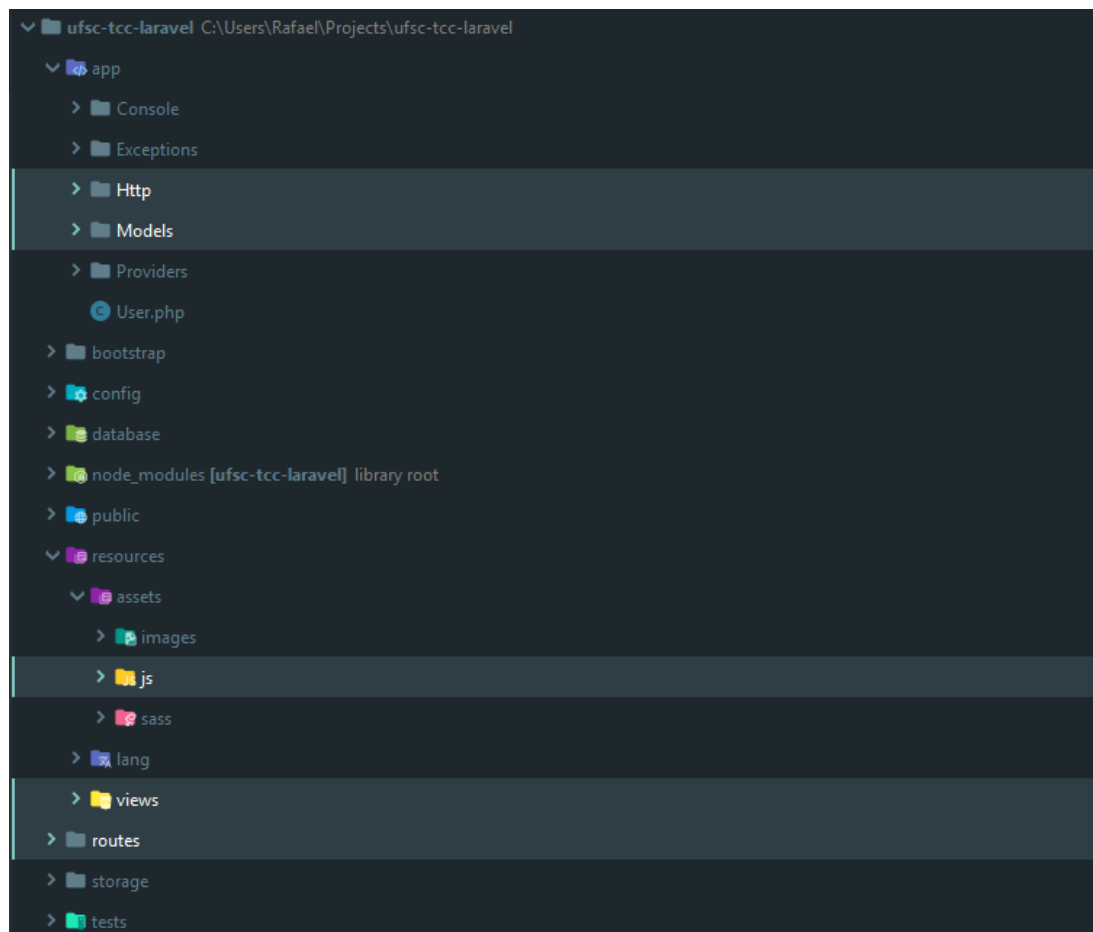
**** Condição que indica que os stakeholders serão consultados indiretamente, de forma individual e nem sempre estarão perto do analista.

As características destacadas (*, **, ***) devem induzir o entrevistado a escolher entre as técnicas Questionários, Laddering ou Repertory grids devido a sua classificação.

APÊNDICE F – Desenvolvimento - Código fonte

Este capítulo apresenta o código fonte da aplicação desenvolvida. Serão apresentados somente os arquivos criados, os arquivos relacionados aos frameworks utilizados não constam neste apêndice. Na **figura 42** são apresentados os diretórios que contém arquivos desenvolvidos (apenas diretórios selecionados). No diretório *resources/assets/js* consta a aplicação frontend que corresponde ao guia propriamente dito e o restante faz parte do backend que fornece a API de acesso e o painel administrativo. A **seção F.1** apresenta o código fonte utilizado na aplicação desenvolvida. O código fonte completo pode ser acessado através do repositório <https://github.com/rafaelign/ufsc-tcc-sin-guia-facetado>.

Figura 42 – Diretórios utilizados no desenvolvimento



Fonte: Elaboração do autor, 2018

F.1 CÓDIGO FONTE

```
<?php
```

```
// app/Http/Controllers/ClassificationController.php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Classification;
use App\Models\FacetReference;
use App\Models\Reference;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
```

```

use Illuminate\Validation\Rule;

class ClassificationController extends Controller
{
    public function index(Request $request)
    {
        if ($request->ajax()) {
            return response()->json(Classification::where('published', 1)
                ->get(['title', 'slug', 'classification_type', 'main_menu']));
        }

        return view('classification.index', [
            'classifications' => Classification::all()
        ]);
    }

    public function edit(int $id)
    {
        $classification = Classification::find($id);

        return view('classification.edit', [
            'classification' => $classification,
            'id' => $id,
        ]);
    }

    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'title' => 'required|max:50',
            'slug' => 'required|unique:classifications|max:50',
            'classification_type' => 'required|max:150',
            'main_menu' => 'required|max:150',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('classifications.edit', ['id' => 0])
                ->withErrors($validator)
                ->withInput();
        }

        $classification = new Classification;

        $classification->title = $request->title;
        $classification->slug = str_slug($request->slug);
        $classification->description = $request->description;
        $classification->classification_type = $request->classification_type;
        $classification->main_menu = $request->main_menu;

        if ($classification->save()) {
            toastr()->success('Cadastro efetuado com sucesso!');

            return redirect()
                ->route('classifications');
        }

        toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais tarde.');
```

```

        return redirect()
            ->route('classifications.edit', ['id' => 0])
            ->withInput();
    }

    public function update(Request $request, int $id)
    {
        $validator = Validator::make($request->all(), [
            'title' => 'required|max:50',
            'slug' => [

```

```

        'required',
        'max:50',
        Rule::unique('classifications')->ignore($id, 'id')
    ],
    'classification_type' => 'required|max:150',
    'main_menu' => 'required|max:150',
]);

if ($validator->fails()) {
    toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

    return redirect()
        ->route('classifications.edit', ['id' => $id])
        ->withErrors($validator)
        ->withInput();
}

$classification = Classification::find($id);

$classification->title = $request->title;
$classification->slug = $request->slug;
$classification->description = $request->description;
$classification->classification_type = $request->classification_type;
$classification->main_menu = $request->main_menu;

if ($classification->save()) {
    toastr()->success('Atualização efetuada com sucesso!');

    return redirect()
        ->route('classifications');
}

toastr()->error('Ocorreu um problema ao gravar as informações,
tente novamente mais tarde.');
```

```

return redirect()
    ->route('classifications.edit', ['id' => $id])
    ->withInput();
}

public function getClassificationBySlug(string $slug)
{
    return response()->json(
        Classification::where('slug', '=', $slug)
            ->where('published', 1)
            ->with('entities')
            ->first()
    );
}

public function getFacetsByClassificationSlug(string $slug)
{
    $collection = Classification::where('slug', '=', $slug)
        ->where('published', 1)
        ->first();

    $facets = $collection ? $collection
        ->facets()
        ->with('values')
        ->get() : [];

    return response()->json($facets);
}

public function getFacetsReferencesByClassificationSlug(string $slug)
{
    $collection = Classification::where('slug', '=', $slug)
        ->where('published', 1)
        ->first();

    $facets = $collection ? $collection
        ->facets()
        ->get(['id']) : [];
}

```

```

        $references = Reference::join('facets_references', 'facets_references.reference_id', '
            references.id')
        ->where(function ($query) use ($facets) {
            $query->whereIn('id', FacetReference::whereIn('facet_id', $facets)
                ->get(['reference_id']));
        })->orderBy('facets_references.code', 'ASC')
        ->groupBy('description', 'facets_references.code')
        ->get(['description', 'facets_references.code']);

        return response()->json($references);
    }

    public function updatePublishedStatus(Request $request, int $id)
    {
        $validator = Validator::make($request->all(), [
            'published' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('classifications.edit', ['id' => 0])
                ->withErrors($validator)
                ->withInput();
        }

        $classification = Classification::find($id);

        if ($classification) {
            $classification->published = (int) $request->published;

            if ($classification->save()) {
                if ((int) $request->published === 1) {
                    toastr()->success('Classificação publicada com sucesso!');
                } else {
                    toastr()->success('Classificação despublicada com sucesso!');
                }

                return response()->json([
                    'error' => false,
                    'message' => 'Classificação atualizada com sucesso!',
                ]);
            }
        }

        if ((int) $request->published === 1) {
            toastr()->error('Ocorreu um erro ao publicar a classificação, tente novamente mais
                tarde.');
```

tarde.');

```

        } else {
            toastr()->error('Ocorreu um erro ao despublicar a classificação, tente novamente
                mais tarde.');
```

mais tarde.');

```

        }

        return response()->json([
            'error' => true,
            'message' => 'Erro ao atualizar classificação',
        ]);
    }
}

<?php

// app/Http/Controllers/EntityController.php

namespace App\Http\Controllers;

use App\Models\Entity;
use App\Models\Facet;
use App\Models\FacetGroup;
use App\Models\Reference;
use App\Models\Value;

```

```

use App\Models\Classification;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\Rule;

class EntityController extends Controller
{
    public function getBySlug(string $slug)
    {
        return response()->json(
            Entity::where('slug', '=', $slug)
                ->where('published', Entity::PUBLISHED)
                ->with('references')
                ->first()
        );
    }

    public function getReferencesByEntitySlug(string $slug)
    {
        $entity = Entity::where('slug', '=', $slug)
            ->where('published', Entity::PUBLISHED)
            ->first();

        $references = Reference::join('entities_references', 'entities_references.reference_id',
            'references.id')
            ->where('entities_references.entity_id', $entity->id)
            ->orderBy('entities_references.code', 'ASC')
            ->groupBy('description', 'entities_references.code')
            ->get(['description', 'entities_references.code']);

        return response()->json($references);
    }

    public function getByClassificationSlug(Request $request, string $slug)
    {
        $entities = Entity::join('classifications', 'classifications.id', 'entities.classification_id')
            ->where('classifications.slug', '=', $slug)
            ->where('entities.published', Entity::PUBLISHED)
            ->with('values');

        if ($request->isMethod('post')) {
            $post = json_decode(file_get_contents('php://input'));

            $mode = isset($post->mode) ? str_slug($post->mode) : 'restrict';
            $filters = isset($post->values) ? $post->values : [];

            foreach ($filters as $key => $filter) {
                if (isset($filter->value) && !$filter->value) {
                    unset($filters[$key]);
                }
            }

            if (!empty($filters)) {
                /**
                 * Para cada filtro selecionado, uma subquery é adicionada
                 * TODO: Refatorar consulta para melhorar performance
                 */
                $values = [];
                foreach ($filters as $filter) {
                    if (is_array($filter->value)) {
                        if ($mode === 'restrict') {
                            foreach ($filter->value as $value) {
                                if (!in_array($value, $values)) {
                                    $values[] = $value;
                                }
                            }
                        } else {
                            $values[] = $filter->value;
                        }
                    } else {
                        if (!in_array($filter->value, $values)) {
                            $values[] = $filter->value;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }

    foreach ($values as $value) {
        $entities->whereIn('entities.id', function ($query) use ($value) {
            if (is_array($value)) {
                $query->select('entity_id')
                    ->from('entities_values')
                    ->whereIn('value_id', $value)
                    ->groupBy('entity_id')->get();
            } else {
                $query->select('entity_id')
                    ->from('entities_values')
                    ->where('value_id', $value)
                    ->groupBy('entity_id')->get();
            }
        });
    }
}

return response()->json(
    $entities->get([
        'entities.id',
        'entities.title',
        'entities.slug',
        'entities.short_description',
    ])
);
}

public function getValuesByEntitySlug(string $slug)
{
    $values = Value::join('entities_values', 'value_id', 'id')
        ->whereIn('entities_values.entity_id', function ($query) use ($slug) {
            $query->select('id')
                ->from('entities')
                ->where('slug', '=', $slug)
                ->where('published', '=', Entity::PUBLISHED)
                ->get(['id']);
        })
        ->get([
            'values.id',
            'values.title',
            'values.value',
            'values.description',
            'values.facet_id',
        ]);

    return response()->json($values);
}

public function addPageView(int $id)
{
    $entity = Entity::find($id);
    $entity->page_views = (int) $entity->page_views + 1;
    $entity->save();

    return response()->json([
        'id' => $entity->id,
        'page_views' => $entity->page_views,
    ]);
}

public function index(int $classificationId)
{
    return view('entity.index', [
        'classification' => Classification::find($classificationId),
        'entities' => Entity::where('classification_id', $classificationId)
            ->paginate(5)
    ]);
}

```

```

public function edit(int $classificationId, int $id)
{
    return view('entity.edit', [
        'classification' => Classification::find($classificationId),
        'entity' => Entity::find($id),
        'classificationId' => $classificationId,
        'id' => $id,
    ]);
}

public function updatePublishedStatus(Request $request, int $classificationId, int $id)
{
    $validator = Validator::make($request->all(), [
        'published' => 'required',
    ]);

    if ($validator->fails()) {
        toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

        return redirect()
            ->route('classifications.entities', ['classificationId' => $classificationId])
            ->withErrors($validator)
            ->withInput();
    }

    $entity = Entity::find($id);

    if ($entity) {
        $entity->published = (int) $request->published;;

        if ($entity->save()) {
            if ((int) $request->published === 1) {
                toastr()->success('Técnica publicada com sucesso!');
            } else {
                toastr()->success('Técnica despublicada com sucesso!');
            }

            return response()->json([
                'error' => false,
                'message' => 'Técnica atualizada com sucesso!',
            ]);
        }
    }

    if ((int) $request->published === 1) {
        toastr()->error('Ocorreu um erro ao publicar a técnica, tente novamente mais tarde
        .');
    } else {
        toastr()->error('Ocorreu um erro ao despublicar a técnica, tente novamente mais
        tarde.');
```



```

        ->route('entities.edit', ['classificationId' => (int) $request->
            classification_id, 'id' => 0])
        ->withErrors($validator)
        ->withInput();
    }

    $entity = new Entity();

    $entity->classification_id = $request->classification_id;
    $entity->title = $request->title;
    $entity->slug = str_slug($request->slug);
    $entity->short_description = $request->short_description;
    $entity->description = $request->description;
    $entity->pros = $request->pros;
    $entity->cons = $request->cons;
    $entity->images = $request->images;
    $entity->published = 0;
    $entity->page_views = 0;
    $entity->user_id = \Auth::user()->id;

    if ($entity->save()) {
        toastr()->success('Cadastro efetuado com sucesso!');

        return redirect()
            ->route('classifications.entities', ['classificationId' => $request->
                classification_id]);
    }

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
        tarde.');
```

```

    return redirect()
        ->route('classifications.entities', ['classificationId' => $request->
            classification_id])
        ->withInput();
}

public function update(Request $request, int $classificationId, int $id)
{
    $validator = Validator::make($request->all(), [
        'title' => 'required|max:100',
        'slug' => [
            'required',
            'max:150',
            Rule::unique('entities')->ignore($id, 'id')
        ],
        'short_description' => 'required|max:250',
        'description' => 'required',
    ]);

    if ($validator->fails()) {
        toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

        return redirect()
            ->route('entities.edit', ['classificationId' => (int) $classificationId, 'id'
                => $id])
            ->withErrors($validator)
            ->withInput();
    }

    $entity = Entity::find($id);

    $entity->title = $request->title;
    $entity->slug = str_slug($request->slug);
    $entity->short_description = $request->short_description;
    $entity->description = $request->description;
    $entity->pros = $request->pros;
    $entity->cons = $request->cons;
    $entity->images = $request->images;

    if ($entity->save()) {
        toastr()->success('Cadastro efetuado com sucesso!');
    }
}

```

```

        return redirect()
            ->route('entities.edit', ['classificationId' => $classificationId, 'id' => $id
                ]);
    }

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
        tarde.');
```

```

    return redirect()
        ->route('classifications.entities', ['classificationId' => $classificationId])
        ->withInput();
}

public function references(Request $request, int $classificationId, int $entityId)
{
    if ($request->isMethod('post')) {
        $validator = Validator::make($request->all(), [
            'entity_id' => 'required',
            'description' => 'required',
            'code' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('entities.references', ['classificationId' => (int)
                    $classificationId, 'entityId' => $entityId])
                ->withErrors($validator)
                ->withInput();
        }

        $entity = Entity::find($entityId);

        if ($entity) {
            $reference = new Reference();
            $reference->description = $request->description;

            if ($reference->save()) {
                toastr()->success('Cadastro efetuado com sucesso!');

                $entity->references()->attach($reference, ['code' => $request->code]);
            }
        }

        toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente
            mais tarde.');
```

```

    return redirect()
        ->route('entities.references', ['classificationId' => $classificationId, '
            entityId' => $entityId]);
}

$entity = Entity::with('references')->find($entityId);

return view('entity.references.index', [
    'entity' => $entity,
    'classificationId' => $classificationId,
    'entityId' => $entityId,
]);
}

public function detachReference(int $entityId, int $id)
{
    $entity = Entity::find($entityId);

    if ($entity) {
        if ($entity->references()->detach($id)) {
            toastr()->success('Registro desvinculado com sucesso!');

            return response()->json([
                'error' => false,
                'message' => 'Referência removida com sucesso!',
            ]);
        }
    }
}

```

```

    });
    }
}

toastr()->error('Ocorreu um erro ao desvincular o registro, tente novamente mais tarde
. ');

return response()->json([
    'error' => true,
    'message' => 'Ocorreu um erro ao remover referência',
]);
}

public function classification(Request $request, int $classificationId, int $entityId)
{
    if ($request->isMethod('post')) {
        $validator = Validator::make($request->all(), [
            'classification_id' => 'required',
            'entity_id' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('entities.classification', ['classificationId' => (int)
                    $classificationId, 'entityId' => $entityId])
                ->withErrors($validator)
                ->withInput();
        }

        $entity = Entity::find($entityId);

        if ($entity) {
            $values = $request->values;

            if (!empty($values)) {
                if ($entity->values()->sync(array_values($values)) ) {
                    toastr()->success('Atualização efetuada com sucesso!');
                }
            }
        }

        toastr()->error('Ocorreu um erro ao gravar as alterações, tente novamente mais
        tarde. ');

        return redirect()
            ->route('entities.classification', ['classificationId' => $classificationId, '
            entityId' => $entityId]);
    }

    $entity = Entity::with('values')->find($entityId);
    $entityValues = [];
    foreach ($entity->values as $v) {
        $entityValues[] = $v->id;
    }
    $facetsGroups = app("App\Http\Controllers\FacetGroupController")->
        getFacetGroupsByClassificationId($classificationId);

    return view('entity.classification.index', [
        'entity' => $entity,
        'entityValues' => $entityValues,
        'facetsGroups' => $facetsGroups,
        'classificationId' => $classificationId,
        'entityId' => $entityId,
    ]);
}
}

<?php

// app/Http/Controllers/FacetController.php

```

```

namespace App\Http\Controllers;

use App\Models\Classification;
use App\Models\Facet;
use App\Models\FacetGroup;
use App\Models\Reference;
use App\Models\Value;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\Rule;

class FacetController extends Controller
{
    public function index(int $classificationId)
    {
        return view('facet.index', [
            'classification' => Classification::find($classificationId),
            'facets' => Facet::where('classification_id', $classificationId)
                ->paginate(5),
            'classificationId' => $classificationId,
        ]);
    }

    public function edit(int $classificationId, int $id)
    {
        return view('facet.edit', [
            'classification' => Classification::find($classificationId),
            'facet' => Facet::find($id),
            'facet_groups' => FacetGroup::orderBy('title')->pluck('title', 'id'),
            'classificationId' => $classificationId,
            'id' => $id,
        ]);
    }

    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'classification_id' => 'required',
            'facet_group_id' => 'required',
            'title' => 'required|max:100',
            'slug' => 'required|unique:entities|max:150',
            'description' => 'required',
            'type' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('facets.edit', ['classificationId' => (int) $request->
                    classification_id, 'id' => 0])
                ->withErrors($validator)
                ->withInput();
        }

        $facet = new Facet();

        $facet->classification_id = $request->classification_id;
        $facet->facet_group_id = $request->facet_group_id;
        $facet->title = $request->title;
        $facet->slug = str_slug($request->slug);
        $facet->description = $request->description;
        $facet->type = $request->type;
        $facet->user_id = \Auth::user()->id;

        if ($facet->save()) {
            toastr()->success('Cadastro efetuado com sucesso!');

            return redirect()
                ->route('classifications.facets', ['classificationId' => $request->
                    classification_id]);
        }
    }
}

```

```

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
    tarde.');
```

```

    return redirect()
    ->route('classifications.facets', ['classificationId' => $request->
        classification_id])
    ->withInput();
}

public function update(Request $request, int $classificationId, int $id)
{
    $validator = Validator::make($request->all(), [
        'facet_group_id' => 'required',
        'title' => 'required|max:100',
        'slug' => [
            'required',
            'max:150',
            Rule::unique('facets')->ignore($id, 'id')
        ],
        'description' => 'required',
        'type' => 'required',
    ]);

    if ($validator->fails()) {
        toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

        return redirect()
            ->route('facets.edit', ['classificationId' => (int) $classificationId, 'id' =>
                $id])
            ->withErrors($validator)
            ->withInput();
    }

    $facet = Facet::find($id);

    $facet->facet_group_id = $request->facet_group_id;
    $facet->title = $request->title;
    $facet->slug = str_slug($request->slug);
    $facet->description = $request->description;
    $facet->type = $request->type;

    if ($facet->save()) {
        toastr()->success('Atualização efetuada com sucesso!');

        return redirect()
            ->route('classifications.facets', ['classificationId' => $classificationId]);
    }

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
    tarde.');
```

```

    return redirect()
    ->route('facets.edit', ['classificationId' => (int) $classificationId, 'id' => $id
        ])
    ->withInput();
}

public function destroy(int $classificationId, int $id)
{
    $facet = Facet::find($id);

    if ($facet) {
        if ($facet->delete()) {
            toastr()->success('Registro removido com sucesso!');

            return response()->json([
                'error' => false,
                'message' => 'Faceta removida',
            ]);
        }
    }

    toastr()->error('Ocorreu um erro ao remover o registro, tente novamente mais tarde.');
```

```

        return response()->json([
            'error' => true,
            'message' => 'Ocorreu um erro ao remover',
        ]);
    }

    public function values(int $classificationId, int $facetId)
    {
        return view('facet.values.index', [
            'facet' => Facet::find($facetId),
            'facetValues' => Value::where(['facet_id' => $facetId])->paginate(5),
            'classificationId' => $classificationId,
            'facetId' => $facetId,
        ]);
    }

    public function editValues(int $classificationId, int $facetId, int $id)
    {
        return view('facet.values.edit', [
            'facet' => Facet::find($facetId),
            'value' => Value::find($id),
            'classificationId' => $classificationId,
            'facetId' => $facetId,
            'id' => $id,
        ]);
    }

    public function storeValue(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'classification_id' => 'required',
            'facet_id' => 'required',
            'title' => 'required|max:100',
            'slug' => 'required|unique:values|max:190',
            'value' => 'required',
            'description' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('facets.edit.values', ['classificationId' => (int) $request->
                    classification_id, 'facetId' => (int) $request->facet_id, 'id' => 0])
                ->withErrors($validator)
                ->withInput();
        }

        $value = new Value();

        $value->facet_id = $request->facet_id;
        $value->title = $request->title;
        $value->slug = str_slug($request->slug);
        $value->value = $request->value;
        $value->description = $request->description;

        if ($value->save()) {
            toastr()->success('Cadastro efetuado com sucesso!');

            return redirect()
                ->route('facets.values', ['classificationId' => $request->classification_id, '
                    facetId' => $request->facet_id]);
        }

        toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
            tarde.');
```

```

        return redirect()
            ->route('facets.values', ['classificationId' => $request->classification_id, '
                facetId' => $request->facet_id])
            ->withInput();
    }
}

```

```

public function updateValue(Request $request, int $classificationId, int $facetId, int $id
)
{
    $validator = Validator::make($request->all(), [
        'classification_id' => 'required',
        'facet_id' => 'required',
        'title' => 'required|max:100',
        'slug' => [
            'required',
            'max:150',
            Rule::unique('values')->ignore($id, 'id')
        ],
        'value' => 'required',
        'description' => 'required',
    ]);

    if ($validator->fails()) {
        toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

        return redirect()
            ->route('facets.edit.values', ['classificationId' => (int) $classificationId,
                'facetId' => $facetId, 'id' => $id])
            ->withErrors($validator)
            ->withInput();
    }

    $value = Value::find($id);

    $value->title = $request->title;
    $value->slug = str_slug($request->slug);
    $value->value = $request->value;
    $value->description = $request->description;

    if ($value->save()) {
        toastr()->success('Atualização efetuada com sucesso!');

        return redirect()
            ->route('facets.values', ['classificationId' => $classificationId, 'facetId'
                => $facetId]);
    }

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
        tarde.');
```

```

    return redirect()
        ->route('facets.edit.values', ['classificationId' => (int) $classificationId, '
            facetId' => $facetId, 'id' => $id])
        ->withInput();
}

public function destroyValue(int $id)
{
    $value = Value::find($id);

    if ($value) {
        if ($value->delete()) {
            toastr()->success('Registro removido com sucesso!');

            return response()->json([
                'error' => false,
                'message' => 'Valor removido com sucesso!',
            ]);
        }
    }

    toastr()->error('Ocorreu um erro ao remover o registro, tente novamente mais tarde.');
```

```

    return response()->json([
        'error' => true,
        'message' => 'Ocorreu um erro ao remover',
    ]);
}

```

```

public function references(Request $request, int $classificationId, int $facetId)
{
    if ($request->isMethod('post')) {
        $validator = Validator::make($request->all(), [
            'facet_id' => 'required',
            'description' => 'required',
            'code' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('facets.references', ['classificationId' => (int)
                    $classificationId, 'facetId' => $facetId])
                ->withErrors($validator)
                ->withInput();
        }

        $facet = Facet::find($facetId);

        if ($facet) {
            $reference = new Reference();
            $reference->description = $request->description;

            if ($reference->save()) {
                toastr()->success('Cadastro efetuado com sucesso!');

                $facet->references()->attach($reference, ['code' => $request->code]);
            }
        }

        toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

        return redirect()
            ->route('facets.references', ['classificationId' => $classificationId, '
                facetId' => $facetId]);
    }

    $facet = Facet::with('references')->find($facetId);

    return view('facet.references.index', [
        'facet' => $facet,
        'classificationId' => $classificationId,
        'facetId' => $facetId,
    ]);
}

public function detachReference(int $facetId, int $id)
{
    $facet = Facet::find($facetId);

    if ($facet) {
        if ($facet->references()->detach($id)) {
            toastr()->success('Registro desvinculado com sucesso!');

            return response()->json([
                'error' => false,
                'message' => 'Referência removida com sucesso!',
            ]);
        }
    }

    toastr()->error('Ocorreu um erro ao desvincular o registro, tente novamente mais tarde
        .');

    return response()->json([
        'error' => true,
        'message' => 'Ocorreu um erro ao remover referência',
    ]);
}
}

```



```

<?php

// app/Http/Controllers/FacetGroupController.php

namespace App\Http\Controllers;

use App\Models\FacetGroup;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\Rule;

class FacetGroupController extends Controller
{
    public function getFacetGroupsByClassificationSlug(string $slug)
    {
        return response()->json(FacetGroup::whereIn('id', function ($query) use ($slug) {
            $query->select('facet_group_id')
                ->from('facets')
                ->join('classifications', 'classifications.id', 'facets.classification_id')
                ->where('classifications.slug', $slug);
        })
            ->orderBy('layout')
            ->with('facets.values')
            ->get());
    }

    public function getFacetGroupsByClassificationId(int $id)
    {
        return FacetGroup::whereIn('id', function ($query) use ($id) {
            $query->select('facet_group_id')
                ->from('facets')
                ->join('classifications', 'classifications.id', 'facets.classification_id')
                ->where('classifications.id', $id);
        })
            ->orderBy('layout')
            ->with('facets.values')
            ->get();
    }

    public function index()
    {
        return view('facet_group.index', [
            'facetsGroups' => FacetGroup::paginate(5)
        ]);
    }

    public function edit(int $id)
    {
        return view('facet_group.edit', [
            'facetGroup' => FacetGroup::find($id),
            'id' => $id,
        ]);
    }

    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'title' => 'required|max:100',
            'layout' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('facets_groups.edit', ['id' => 0])
                ->withErrors($validator)
                ->withInput();
        }

        $facetGroup = new FacetGroup();

        $facetGroup->title = $request->title;
    }
}

```

```

        $facetGroup->layout = $request->layout;

        if ($facetGroup->save()) {
            toastr()->success('Cadastro efetuado com sucesso!');

            return redirect()
                ->route('facets_groups');
        }

        toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
            tarde.');
```

```

        return redirect()
            ->route('facets_groups')
            ->withInput();
    }

    public function update(Request $request, int $id)
    {
        $validator = Validator::make($request->all(), [
            'title' => 'required|max:100',
            'layout' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()
                ->route('facets_groups.edit', ['id' => $id])
                ->withErrors($validator)
                ->withInput();
        }

        $facetGroup = FacetGroup::find($id);

        $facetGroup->title = $request->title;
        $facetGroup->layout = $request->layout;

        if ($facetGroup->save()) {
            toastr()->success('Atualização efetuada com sucesso!');

            return redirect()
                ->route('facets_groups');
        }

        toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
            tarde.');
```

```

        return redirect()
            ->route('facets_groups.edit', ['id' => $id])
            ->withInput();
    }
}

<?php
// app/Http/Controllers/HomeController.php

namespace App\Http\Controllers;

use App\Models\Entity;
use Illuminate\Http\Request;

class HomeController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function index()
    {
        $accesses = Entity::select(\DB::raw('sum(page_views) as total'))->first();
    }
}

```

```

        $pages = Entity::orderBy('page_views')->first(['title', 'page_views']);

        return view('home', [
            'total_accesses' => $accesses->total,
            'popular_page' => $pages->title,
        ]);
    }
}

<?php

// app/Http/Controllers/UserController.php

namespace App\Http\Controllers;

use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class UserController extends Controller
{
    public function index()
    {
        return view('user.index', [
            'users' => User::paginate(15),
        ]);
    }

    public function edit(int $id)
    {
        $user = User::find($id);

        return view('user.edit', [
            'user' => $user,
            'id' => $id,
        ]);
    }

    public function destroy(int $id)
    {
        $user = User::find($id);

        if ($user) {
            if ($user->delete()) {
                toastr()->success('Registro removido com sucesso!');

                return response()->json([
                    'error' => false,
                    'message' => 'User deleted',
                ]);
            }
        }

        toastr()->error('Ocorreu um erro ao remover o registro, tente novamente mais tarde.');
```

```

        return response()->json([
            'error' => true,
            'message' => 'User cannot be deleted',
        ]);
    }

    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required|max:255',
            'email' => 'required|unique:users|max:255',
            'password' => 'required',
        ]);

        if ($validator->fails()) {
            toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

            return redirect()

```

```

        ->route('users.edit', ['id' => 0])
        ->withErrors($validator)
        ->withInput();
    }

    $user = new User;

    $user->name = $request->name;
    $user->email = $request->email;
    $user->password = bcrypt($request->password);

    if ($user->save()) {
        toastr()->success('Cadastro efetuado com sucesso!');

        return redirect()
            ->route('users');
    }

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
        tarde.');
```

```

    return redirect()
        ->route('users.edit', ['id' => 0])
        ->withInput();
}

public function update(Request $request, int $id)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required|max:255',
        'email' => 'required|unique:users|max:255',
    ]);

    if ($validator->fails()) {
        toastr()->error('Informações inválidas. Verifica as informações fornecidas!');

        return redirect()
            ->route('users.edit', ['id' => $id])
            ->withErrors($validator)
            ->withInput();
    }

    $user = User::find($id);

    $user->name = $request->name;
    $user->email = $request->email;

    if ($user->save()) {
        toastr()->success('Atualização efetuada com sucesso!');

        return redirect()
            ->route('users');
    }

    toastr()->error('Ocorreu um problema ao gravar as informações, tente novamente mais
        tarde.');
```

```

    return redirect()
        ->route('users.edit', ['id' => $id])
        ->withInput();
    }
}

<?php

// app/Http/Controllers/VueController.php

namespace App\Http\Controllers;

use App\Models\Entity;
use Illuminate\Http\Request;
use JavaScript;
```

```

class VueController extends Controller
{
    public function index()
    {
        JavaScript::put([
            'url' => config('app.url')
        ]);

        return view('vue.index');
    }
}

<?php

// app/Models/Classification.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Classification extends Model
{
    protected $table = 'classifications';

    protected $fillable = [
        'title',
        'slug',
    ];

    public function facets()
    {
        return $this->hasMany(Facet::class);
    }

    public function entities()
    {
        return $this->hasMany(Entity::class)->where('published', 1);
    }
}

<?php

// app/Models/Entity.php

namespace App\Models;

use App\User;
use Illuminate\Database\Eloquent\Model;

class Entity extends Model
{
    /** @var int PUBLISHED */
    const PUBLISHED = 1;

    protected $table = 'entities';

    protected $fillable = [
        'title',
        'short_description',
        'description',
        'pros',
        'cons',
        'images',
        'published',
    ];

    protected $appends = ['images_array'];

    public function values()
    {
        return $this->belongsToMany(Value::class, 'entities_values', 'entity_id');
    }
}

```

```

    public function classification()
    {
        return $this->belongsTo(Classification::class);
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function references()
    {
        return $this->belongsToMany(Reference::class, 'entities_references', 'entity_id')->
            withPivot('code');
    }

    public function getImagesArrayAttribute()
    {
        return !is_null($this->images) ? json_decode($this->images) : null;
    }
}

<?php
// app/Models/EntityReference.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class EntityReference extends Model
{
    protected $table = 'entities_references';

    protected $fillable = [
        'entity_id',
        'reference_id',
        'code',
    ];

    public function reference()
    {
        return $this->belongsTo(Reference::class);
    }
}

<?php
// app/Models/EntityValue.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class EntityValue extends Model
{
    protected $table = 'entities_values';

    protected $fillable = [
        'entity_id',
        'value_id',
    ];

    public function value()
    {
        return $this->belongsTo(Value::class);
    }

    public function entity()
    {
        return $this->belongsTo(Entity::class);
    }
}

```

```

<?php

// app/Models/Facet.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Facet extends Model
{
    protected $table = 'facets';

    protected $fillable = [
        'title',
        'slug',
        'description',
        'type',
    ];

    public function classification()
    {
        return $this->belongsTo(Classification::class);
    }

    public function references()
    {
        return $this->belongsToMany(Reference::class, 'facets_references', 'facet_id')->
            withPivot('code');
    }

    public function values()
    {
        return $this->hasMany(Value::class);
    }

    public function group()
    {
        return $this->belongsTo(FacetGroup::class, 'facet_group_id', 'id');
    }
}

<?php

// app/Models/FacetGroup.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class FacetGroup extends Model
{
    protected $table = 'facet_groups';

    protected $fillable = [
        'title',
    ];

    public function facets()
    {
        return $this->hasMany(Facet::class);
    }
}

<?php

// app/Models/FacetReference.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class FacetReference extends Model
{

```

```

        protected $table = 'facets_references';

        protected $fillable = [
            'facet_id',
            'reference_id',
            'code',
        ];

        public function reference()
        {
            return $this->belongsTo(Reference::class);
        }
    }

<?php
// app/Models/Reference.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Reference extends Model
{
    protected $table = 'references';

    protected $fillable = [
        'description',
    ];

    public function entities()
    {
        return $this->belongsToMany(Entity::class);
    }

    public function facets()
    {
        return $this->belongsToMany(Facet::class);
    }

    public function facets_references()
    {
        return $this->hasMany(FacetReference::class);
    }
}

<?php
// app/Models/Value.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Value extends Model
{
    protected $table = 'values';

    protected $fillable = [
        'title',
        'slug',
        'value',
        'type',
    ];

    public function facet()
    {
        return $this->belongsTo(Facet::class);
    }

    public function entity()
    {
        return $this->belongsToMany(Entity::class, 'entities_values', 'value_id');
    }
}

```



```

    }
}

<?php
// routes/api.php

use Illuminate\Http\Request;

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('classifications', 'ClassificationController@index');

Route::get('classifications/{slug}', 'ClassificationController@getClassificationBySlug');

Route::get('classifications/{slug}/entities', 'EntityController@getByClassificationSlug');

Route::post('classifications/{slug}/entities', 'EntityController@getByClassificationSlug');

Route::get('classifications/{slug}/facets', '
    ClassificationController@getFacetsByClassificationSlug');

Route::get('classifications/{slug}/facets/references', '
    ClassificationController@getFacetsReferencesByClassificationSlug');

Route::get('entities/{slug}', 'EntityController@getBySlug');

Route::get('entities/{slug}/references', 'EntityController@getReferencesByEntitySlug');

Route::get('entities/{slug}/values', 'EntityController@getValuesByEntitySlug');

Route::get('entities/page_views/{id}', 'EntityController@addPageView');

Route::get('facet_groups/{classificationSlug}', '
    FacetGroupController@getFacetGroupsByClassificationSlug');

<?php
// routes/web.php

Route::get('/', function () { return redirect('/app'); });

Route::get('/app/{vue_capture?}', 'VueController@index')->where('vue_capture', '[\\/\w\.-]*');

Route::prefix('admin')->group(function () {
    Route::get('/', 'HomeController@index')->name('home');

    // Classifications
    Route::get('/classifications', 'ClassificationController@index')->name('classifications');
    Route::get('/classifications/{id}', 'ClassificationController@edit')->name('
        classifications.edit');
    Route::post('/classifications', 'ClassificationController@store')->name('classifications.
        store');
    Route::put('/classifications/{id}', 'ClassificationController@update')->name('
        classifications.update');
    Route::put('/classifications/{id}/update_publish', '
        ClassificationController@updatePublishedStatus')->name('classifications.publish');

    // Entities
    Route::get('/classifications/{classificationId}/entities/{id}', 'EntityController@edit')->
        name('entities.edit');
    Route::get('/classifications/{classificationId}/entities', 'EntityController@index')->name(
        'classifications.entities');
    Route::post('/classifications/{classificationId}/entities', 'EntityController@store')->
        name('entities.store');
    Route::put('/classifications/{classificationId}/entities/{id}', 'EntityController@update')
        ->name('entities.update');
    Route::put('/classifications/{classificationId}/entities/{id}/update_publish', '
        EntityController@updatePublishedStatus')->name('entities.publish');

```

```

Route::get('/classifications/{classificationId}/entities/{entityId}/classification', '
    EntityController@classification')->name('entities.classification');
Route::post('/classifications/{classificationId}/entities/{entityId}/classification', '
    EntityController@classification')->name('entities.classification.post');

// Facets References
Route::get('/classifications/{classificationId}/entities/{entityId}/references', '
    EntityController@references')->name('entities.references');
Route::post('/classifications/{classificationId}/entities/{entityId}/references', '
    EntityController@references')->name('entities.references.post');
Route::delete('/entities/{entityId}/references/{id}', 'EntityController@detachReference')
    ->name('entities.detach.reference');

// Facets
Route::get('/classifications/{classificationId}/facets/{id}', 'FacetController@edit')->
    name('facets.edit');
Route::get('/classifications/{classificationId}/facets', 'FacetController@index')->name('
    classifications.facets');
Route::post('/classifications/{classificationId}/facets', 'FacetController@store')->name('
    facets.store');
Route::put('/classifications/{classificationId}/facets/{id}', 'FacetController@update')->
    name('facets.update');
Route::delete('/classifications/{classificationId}/facets/{id}', 'FacetController@destroy'
    )->name('users.delete');

// Facets Values
Route::get('/classifications/{classificationId}/facets/{facetId}/values', '
    FacetController@values')->name('facets.values');
Route::get('/classifications/{classificationId}/facets/{facetId}/values/{id}', '
    FacetController@editValue')->name('facets.edit.values');
Route::post('/classifications/{classificationId}/facets/{facetId}/values', '
    FacetController@storeValue')->name('facets.store.values');
Route::put('/classifications/{classificationId}/facets/{facetId}/values/{id}', '
    FacetController@updateValue')->name('facets.update.values');
Route::delete('/values/{id}', 'FacetController@destroyValue')->name('facets.destroy.values
    ');

// Facets References
Route::get('/classifications/{classificationId}/facets/{facetId}/references', '
    FacetController@references')->name('facets.references');
Route::post('/classifications/{classificationId}/facets/{facetId}/references', '
    FacetController@references')->name('facets.references.post');
Route::delete('/facets/{facetId}/references/{id}', 'FacetController@detachReference')->
    name('facets.detach.reference');

http://localhost:8080/admin/facets/1/reference/43

// Facets Groups
Route::get('/facets_groups', 'FacetGroupController@index')->name('facets_groups');
Route::get('/facets_groups/{id}', 'FacetGroupController@edit')->name('facets_groups.edit')
    ;
Route::post('/facets_groups', 'FacetGroupController@store')->name('facets_groups.store');
Route::put('/facets_groups/{id}', 'FacetGroupController@update')->name('facets_groups.
    update');

// Users
Route::get('/users', 'UserController@index')->name('users');
Route::get('/users/{id}', 'UserController@edit')->name('users.edit');
Route::post('/users', 'UserController@store')->name('users.store');
Route::put('/users/{id}', 'UserController@update')->name('users.update');
Route::delete('/users/{id}', 'UserController@destroy')->name('users.delete');

// Authentication Routes...
Route::get('login', 'Auth\LoginController@showLoginForm')->name('login');
Route::post('login', 'Auth\LoginController@login');
Route::post('logout', 'Auth\LoginController@logout')->name('logout');

// Password Reset Routes...
Route::get('password/reset', 'Auth\ForgotPasswordController@showLinkRequestForm')->name('
    password.request');
Route::post('password/email', 'Auth\ForgotPasswordController@sendResetLinkEmail')->name('
    password.email');
Route::get('password/reset/{token}', 'Auth\ResetPasswordController@showResetForm');

```

```

Route::post('password/reset', 'Auth\ResetPasswordController@reset')->name('password.reset'
    );
});

<?php

// resources/views/classification/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ __('Classificações') }}</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <h2 class="subtitle">
        <a href="{ route('classifications.edit', ['id' => 0]) }}" class="button is-medium is-pulled-right">
            <span class="icon"><span class="mdi mdi-check"></span></span> <span>{{ __('Nova classificação') }}</span>
        </a>

        {{ __('Classificações') }}
    </h2>
    <section class="info-tiles">
        @foreach ($classifications as $classification)
            <div class="card">
                <header class="card-header">
                    <p class="card-header-title">{{ $classification->title }}</p>
                </header>
                <div class="card-content">
                    <div class="content">
                        {{ $classification->description }}
                    </div>
                </div>
                <footer class="card-footer">
                    <a href="{ route('classifications.edit', ['id' => $classification->id]) }}" class="card-footer-item">
                        <span class="icon"><span class="mdi mdi-pencil"></span></span> {{ __('Editar') }}
                    </a>
                    <a href="{ route('classifications.entities', ['classificationId' => $classification->id]) }}" class="card-footer-item">
                        <span class="icon"><span class="mdi mdi-application"></span></span> <span class="badge is-badge-info" data-badge="{ count($classification->entities) }}">{{ __('Técnicas') }}</span>
                    </a>
                    <a href="{ route('classifications.facets', ['classificationId' => $classification->id]) }}" class="card-footer-item">
                        <span class="icon"><span class="mdi mdi-adjust"></span></span> <span class="badge is-badge-info" data-badge="{ count($classification->facets) }}">{{ __('Facetas') }}</span>
                    </a>
                    @if ($classification->published)
                        <a href="#" @click="unpublishClassification({{ $classification->id }}" class="card-footer-item">
                            <span class="icon" v-if="isLoading()"><span class="mdi mdi-spin mdi-loading"></span></span>
                            <span class="icon" v-if="! isLoading()"><span class="mdi mdi-arrow-expand-down"></span></span> {{ __('Despublicar') }}
                        </a>
                    @else
                        <a href="#" @click="publishClassification({{ $classification->id }}" class="card-footer-item">
                            <span class="icon" v-if="isLoading()"><span class="mdi mdi-spin mdi-loading"></span></span>

```

```

        <span class="icon" v-if="! isLoading()"><span class="mdi mdi-arrow
            -expand-up is-success"></span></span> {{ __( 'Publicar' ) }}
        </a>
    @endif
</footer>
</div>
@empty
    <div class="notification is-info">{{ __( 'Nenhuma classificação encontrada.' ) }}</div>
@endforelse
</section>
</endsection>

// resources/assets/js/utils/routes.js

import AboutPage from '../pages/AboutPage.vue';
import CollectionPage from '../pages/ClassificationPage.vue';
import EntityPage from '../pages/EntityPage.vue';
import EntitiesPage from '../pages/EntitiesPage.vue';
import FacetsPage from '../pages/FacetsPage.vue';

export const routes = [
    { path: '/app', component: AboutPage, name: 'about' },
    { path: '/app/classificacoes/:classification', component: CollectionPage, name: 'classificacoes' },
    { path: '/app/classificacoes/:classification/facetass', component: FacetsPage, name: 'classificacoes_facetas' },
    { path: '/app/classificacoes/:classification/entidades', component: EntitiesPage, name: 'classificacoes_entidades' },
    { path: '/app/classificacoes/:classification/entidades/:entity', component: EntityPage, name: 'classificacoes_entidade' }
];

// resources/assets/js/components/Breadcrumb.js

<script>
    export default {
        data () {
            return {}
        },
        props: {
            items: {
                type: Array,
                default: function () {
                    return []
                }
            }
        }
    }
</script>

<template>
    <nav class="breadcrumb has-arrow-separator is-right" aria-label="breadcrumbs">
        <ul>
            <li v-for="item in items" :class="item.active ? 'is-active' : ''">
                <router-link :to="{ path: item.url }" v-if="item.destination === undefined">{{ item.title }}</router-link>
                <router-link :to="item.destination" v-else>{{ item.title }}</router-link>
            </li>
        </ul>
    </nav>
</template>

// resources/assets/js/components/Card.js

<script>
    export default {
        name: 'card',
        props: {
            title: {
                type: String,
                default: null,
                required: true
            }
        }
    }

```

```

    },
    content: {
      type: String,
      default: null,
      required: true
    },
    action: {
      type: String,
      default: null
    }
  },
  data: function () {
    return {}
  }
}
</script>

<template>
  <div class="card">
    <header class="card-header">
      <p class="card-header-title">
        {{ title }}
      </p>
    </header>
    <div class="card-content">
      <div class="content">
        {{ content }}
      </div>
    </div>
    <div class="card-footer" v-if="action">
      <router-link class="card-footer-item" :to="{ path: action }">
        <b-icon icon="link"></b-icon> <span>Acessar</span>
      </router-link>
    </div>
  </div>
</template>

// resources/assets/js/components/Classification.js

<script>
  export default {
    data () {
      return {
        presented: []
      }
    },
    props: {
      items: {
        type: Array,
        default: function () {
          return []
        }
      },
      values: {
        type: Array,
        default: function () {
          return []
        }
      }
    },
    methods: {
      getValue: function (id) {
        return this.values.filter((item) => {
          return item.facet_id === id
        });
      }
    }
  }
}
</script>

<template>
  <section>
    <h2 class="subtitle"><b-icon icon="table" class="has-text-success"></b-icon> <span>

```

```

        Classificação completa</span></h2>
<section v-for="item in items">
  <div class="row">
    <div class="columns is-multiline">
      <div class="column is-12 has-background-grey-lighter has-text-black margin-top-15 padding-5">
        <small>{{ item.title }}</small>
      </div>
      <div class="column is-3" v-for="facet in item.facets">
        <small><b>{{ facet.title }}</b></small>
        <div>
          <div v-if="getValue(facet.id).length">
            <span v-for="value in getValue(facet.id)" class="tag is-primary margin-top-10 margin-right-5">
              <span>{{ value.title }}</span>
            </span>
          </div>
          <span v-else class="tag is-primary margin-top-10">
            <i>null</i>
          </span>
        </div>
      </div>
    </div>
  </div>
</section>
</section>
</template>

// resources/assets/js/components/DynamicField.js

<script>
import { mapMutations } from 'vuex'

export default {
  props: {
    type: {
      type: String
    },
    options: {
      type: Array,
      default: () => {
        return [];
      }
    },
    option: {
      type: String,
      default: null
    },
    name: {
      type: String,
      default: null
    },
  },
  data () {
    return {
      fieldValue: null,
      fieldName: null,
      switchConfig: {
        trueValue: 1,
        falseValue: 0
      },
      sliderConfig: {
        step: 1,
        min: 0,
        max: 2
      }
    }
  },
  created () {
    if (this.type === 'switch') {
      this.switchConfig.falseValue = parseInt(this.options[0].id)
      this.switchConfig.trueValue = parseInt(this.options[1].id)
    }
  }
}

```

```

        this.addDefaultValue({
          name: this.name,
          value: this.switchConfig.falseValue
        })
      }

      if (this.type === 'slider') {
        this.sliderConfig.min = parseInt(this.options[0].value)
        this.sliderConfig.max = parseInt(this.options[ this.options.length - 1 ].value)
      }

      this.addDefaultValue({
        name: this.name,
        value: this.sliderConfig.min
      })
    }

    if (this.type === 'checkbox' || this.type === 'checkboxbutton') {
      this.addDefaultValue({
        name: this.name,
        value: []
      })
    }

    this.fieldName = this.name
    this.fieldValue = this.getFilter(this.name)
  },
  methods: {
    ...mapMutations([
      'addFilter',
      'addDefaultValue'
    ]),
    setFilter () {
      let filter = {
        name: this.fieldName,
        value: this.fieldValue
      }

      if (this.type === 'slider') {
        filter.value = this.options[ this.fieldValue ].id
      }

      this.addFilter(filter)
    },
    switchValueToTitle() {
      let opt = this.options.filter((item) => {
        return item.id === this.fieldValue
      })

      return opt[0].title || ''
    },
    getFilter: function (name) {
      return this.$store.getters.getFilterValueByName(name)
    }
  }
}
</script>

<template>
  <section v-if="type === 'select'">
    <b-select
      placeholder="Selecione uma opção"
      size="is-medium"
      v-model="fieldValue"
      @input="setFilter">
      <option
        v-for="option in options"
        :value="option.id"
        :key="option.id">
        {{ option.title }}
      </option>
    </b-select>
  </section>

```

```

<section class="margin-top-10" v-else-if="type === 'switch'">
  <b-switch
    v-model="fieldValue"
    :true-value="switchConfig.trueValue"
    :false-value="switchConfig.falseValue"
    size="is-medium"
    @input="setFilter">
    {{ switchValueToTitle() }}
  </b-switch>
</section>

<section v-else-if="type === 'slider'">
  <input
    class="slider is-fullwidth is-medium is-primary is-circle has-output"
    :step="sliderConfig.step"
    :min="sliderConfig.min"
    :max="sliderConfig.max"
    v-model="fieldValue"
    :id="name"
    type="range"
    @input="setFilter">
  <output :for="name">{{ options[ fieldValue ].title }}</output>
</section>

<section
  class="margin-top-10"
  v-else-if="type === 'checkbox'">
  <div class="field" v-for="item in options" :key="item.id">
    <b-checkbox
      v-model="fieldValue"
      :native-value="item.id"
      :name="item.slug"
      size="is-medium"
      @input="setFilter">{{ item.title }}</b-checkbox>
    </div>
</section>

<b-field
  class="margin-top-10"
  v-else-if="type === 'checkboxbutton'">
  <b-checkbox-button
    v-for="item in options"
    :key="item.id"
    v-model="fieldValue"
    :native-value="item.id"
    type="is-primary"
    @input="setFilter">
    <span>{{ item.title }}</span>
  </b-checkbox-button>
</b-field>
</template>

// resources/assets/js/components/ModalForm.js

<script>
import { mapMutations } from 'vuex'
import DynamicField from './DynamicField'
import Popover from './Popover'

export default {
  components: {
    DynamicField,
    Popover
  },
  props: {
    title: {
      type: String,
      default: null
    },
    horizontalData: {
      type: Array,
      default: function () {

```



```

        return [];
    }
},
verticalData: {
    type: Array,
    default: function () {
        return [];
    }
},
data: function () {
    return {
        selectedFilters: [],
        mode: 'restrict'
    }
},
methods: {
    ...mapMutations([
        'updateMode'
    ]),
    filter: function () {
        this.$emit('filter')
        this.$parent.close()
    },
    reset: function () {
        this.$emit('reset')
        this.$parent.close()
    },
    setMode: function () {
        this.updateMode(this.mode)
    },
    getMode: function () {
        return this.$store.getters.getMode
    }
},
created () {
    this.mode = this.getMode()
}
}
</script>

<style type="text/css">
    .modal-close { background-color: #BBB; }
</style>

<template>
    <div class="modal-content modal-card">
        <header class="modal-card-head">
            <p class="modal-card-title">{{ title }}</p>
        </header>
        <section class="modal-card-body">
            <div class="has-text-right">
                <b-switch
                    v-model="mode"
                    :true-value="'restrict'"
                    :false-value="'open'"
                    size="is-medium"
                    @input="setMode">
                    Modo restrito
                </b-switch>
                <br>
                <span class="has-background-grey-lighter" v-if="mode === 'restrict'" style="
                    max-width: 590px; display: inline-block; margin-top: 10px;"><b-icon icon="
                    asterisk"></b-icon> Neste modo de busca, para que uma técnica seja
                    apresentada, ela precisa estar associada com <b>todas</b> as
                    características <b>selecionadas</b> de cada faceta <b>utilizada</b>. <br>
                    <small>Somente as facetas utilizadas são consideradas, não é necessário
                    utilizar todas disponíveis.</small></span>
                <span class="has-background-grey-lighter" v-else style="max-width: 655px;
                    display: inline-block; margin-top: 10px;"><b-icon icon="asterisk"></b-icon>
                    Com o modo restrito desativado, para que uma técnica seja apresentada,
                    basta que <b>uma das características</b> selecionadas estejam associadas a
                    ela para cada faceta <b>utilizada</b>. <br> <small>Somente as facetas

```

```

        utilizadas são consideradas, não é necessário utilizar todas disponíveis
        .</small></span>
    </div>
    <form action="">
        <div class="tile is-ancestor">
            <div class="tile is-vertical is-12">
                <div class="tile is-parent" v-for="group in horizontalData">
                    <article class="tile is-child is-white">
                        <p class="subtitle has-text-weight-bold">{{ group.title }}</p>
                        <div class="content">
                            <div class="columns">
                                <div class="column is-3" v-for="facet in group.facets"
                                    >
                                    <div class="field">
                                        <label :for="facet.slug" class="">
                                            <span class="columns">
                                                <span class="column is-three-fifths">
                                                    {{ facet.title }}
                                                    <popover class="is-pulled-right"
                                                        direction="is-popover-bottom">
                                                        <span class="icon is-popover-
                                                            trigger" slot="button"
                                                            slot-scope="{ toggle }"
                                                            @click="toggle()"><span
                                                                class="mdi mdi-information
                                                                mdi-circle"></span></span>
                                                        <div slot="content">
                                                            <p>{{ facet.description
                                                                }}</p>
                                                            <p>Opções:</p>
                                                            <ul>
                                                                <li v-for="value in
                                                                    facet.values">{{
                                                                        value.title }}: {{
                                                                        value.description
                                                                        }}</li>
                                                                </ul>
                                                            </div>
                                                        </popover>
                                                    </span>
                                                </span>
                                            </label>
                                            <dynamic-field
                                                :type="facet.type"
                                                :name="facet.slug"
                                                :options="facet.values"></dynamic-
                                                field>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </article>
                    </div>
                </div>
            <div class="tile">
                <div class="tile is-parent" v-for="group in verticalData">
                    <article class="tile is-child is-white">
                        <p class="subtitle has-text-weight-bold">{{ group.title
                            }}</p>
                        <div class="content">
                            <div class="field" v-for="facet in group.facets">
                                <small>
                                    {{ facet.title }}
                                </small>
                                <popover direction="is-popover-top">
                                    <span class="icon is-popover-trigger" slot
                                        ="button" slot-scope="{ toggle }"
                                        @click="toggle()"><span class="mdi mdi
                                        -information mdi-circle"></span></span>
                                    <div slot="content">
                                        <p>{{ facet.description }}</p>
                                    </div>
                                </popover>
                            </div>
                        </div>
                    </article>
                </div>
            </div>
        </div>
    </form>

```

```

        <p>Opções:</p>
        <ul>
          <li v-for="value in facet.values"
            >{{ value.title }}: {{ value.
              description }}</li>
        </ul>
      </div>
    </popover>
  </small>
  <br>
  <dynamic-field
    :type="facet.type"
    :name="facet.slug"
    :options="facet.values"></dynamic-field>
  </div>
</div>
</article>
</div>
</div>
</div>
</div>
</form>
</section>
<footer class="modal-card-foot buttons is-centered">
  <button class="button is-medium" type="button" @click="reset">
    <b-icon icon="eraser"></b-icon> <span>Limpar</span>
  </button>
  <button class="button is-medium is-primary" @click="filter">
    <b-icon icon="filter"></b-icon> <span>Filtrar</span>
  </button>
</footer>
</div>
</template>

// resources/assets/js/components/Popover.js

<script>
  export default {
    props: {
      direction: {
        type: String,
        default: 'is-popover-bottom'
      }
    },
    data() {
      return {
        active: ''
      }
    },
    methods: {
      toggle: function () {
        this.active = this.active === 'is-popover-active' ? '' : 'is-popover-active'
      }
    }
  }
</script>

<template>
  <div :class="['popover', direction, active]">
    <slot name="button" :toggle="toggle"></slot>
    <div class="popover-content">
      <slot name="content"></slot>
    </div>
  </div>
</template>

// resources/assets/js/components/References.js

<script>
  export default {
    props: {
      title: {
        type: String

```

```

    },
    items: {
      type: Array,
      default: function () {
        return [];
      }
    }
  }
}
</script>

<template>
  <section v-if="items.length">
    <h2 class="subtitle"><b-icon icon="view-list" class="has-text-danger"></b-icon> <span>
      <{{ title }}</span></h2>
    <ul class="references-list">
      <li v-for="(reference, index) in items" :key="index">
        <sup>[{{ reference.code }}]</sup> {{ reference.description }}
      </li>
    </ul>
  </section>
</template>

<style type="text/css">
  ul.references-list {
    padding: 0;
    margin: 0;
  }

  ul.references-list li {
    list-style: none;
    margin-top: 20px;
  }
</style>

// resources/assets/js/components/ShareButtons.js

<script>
import VueGoodshareFacebook from 'vue-goodshare/src/providers/Facebook'
import VueGoodshareGoogleplus from 'vue-goodshare/src/providers/GooglePlus'
import VueGoodshareTwitter from 'vue-goodshare/src/providers/Twitter'
import VueGoodshareLinkedin from 'vue-goodshare/src/providers/LinkedIn'

export default {
  components: {
    VueGoodshareFacebook,
    VueGoodshareGoogleplus,
    VueGoodshareTwitter,
    VueGoodshareLinkedin
  },
  data () {
    return {}
  },
  props: {
    label: {
      type: String,
      default: null
    }
  }
}
</script>

<template>
  <ul class="socialbar">
    <li v-if="label" class="menu-label">
      {{ label }}
    </li>
    <li>
      <vue-goodshare-facebook :page_url="$root.url" has_icon></vue-goodshare-facebook>
      <vue-goodshare-twitter :page_url="$root.url" has_icon></vue-goodshare-twitter>
      <vue-goodshare-linkedin :page_url="$root.url" has_icon></vue-goodshare-linkedin>
    </li>
  </ul>

```

```

</template>

// resources/assets/js/components/Sidebar.js

<script>
import ClipLoader from 'vue-spinner/src/ClipLoader'
import ShareButtons from './ShareButtons'

export default {
  components: {
    ClipLoader,
    ShareButtons
  },
  data: function () {
    return {
      entities: [],
      errors: [],
      isLoadingMenu: false
    }
  },
  created () {
    this.isLoadingMenu = true;

    axios.get(this.$store.getters.getUrl + '/api/classifications')
      .then((response) => {
        this.entities = response.data
        this.isLoadingMenu = false
      })
      .catch((error) => this.errors = error.response.data.errors)
  }
}
</script>

<template>
<aside class="column is-4 is-4-tablet is-3-desktop is-3-widescreen is-2-fullhd is-
fullheight is-narrow-mobile">
<div class="has-text-centered-mobile">
<figure class="image">

</figure>
</div>
<ul class="menu-list">
<li>
<router-link to="/app">
  Sobre este guia
</router-link>
</li>
<li v-for="item,key in entities">
<router-link :to="{ path: '/app/classificacoes/' + item.slug }">
  {{ item.title }}
</router-link>

<ul>
<li>
<router-link :to="{ path: '/app/classificacoes/' + item.slug + '/'
entidades' }">
<b-icon icon="view-list" size="is-small"></b-icon> <span>{{ item.
main_menu }}</span>
</router-link>
</li>
<li>
<router-link :to="{ path: '/app/classificacoes/' + item.slug + '/'
facetas' }">
<b-icon icon="information" size="is-small"></b-icon> <span>Facetas
de Classificação</span>
</router-link>
</li>
</ul>
</li>
<li class="is-hidden-tablet">
<router-link to="#">Compartilhe</router-link>

```

```

        <share-buttons></share-buttons>
      </li>
    </ul>
    <clip-loader v-if="isLoadingMenu" class="is-centered"></clip-loader>

    <nav class="navbar is-fixed-bottom socialbar is-hidden-mobile" style="opacity: unset">
      <div class="navbar-item">
        <share-buttons label="Compartilhe"></share-buttons>
      </div>
    </nav>
  </aside>
</template>

// resources/assets/js/pages/AboutPage.js

<script>
  import Breadcrumb from '../components/Breadcrumb'
  import References from '../components/References'

  export default {
    components: {
      Breadcrumb,
      References
    },
    data () {
      return {}
    }
  }
</script>
<template>
  <section v-if="this.$store.getters.isLoaded">
    <section class="hero">
      <div class="hero-body">
        <breadcrumb :items="[
          { url: '#', title: 'Guia Facetado de Engenharia de Requisitos', active:
            true },
        ]"></breadcrumb>

        <section class="content">
          <h1 class="title"><b-icon icon="information"></b-icon> Sobre este guia</h1>
          <p class="has-text-justified">
            Este guia procura apresentar técnicas de elicitação de requisitos e
            suas respectivas informações de uma maneira didática e simples.
            A execução da pesquisa demonstrou que existem diversos métodos que
            podem ser utilizados para elicitar requisitos de software.
            Outra constatação é que o processo de elicitação de requisitos de um
            projeto não se restringe a utilização de apenas uma técnica, mas
            sim de uma combinação entre elas.
            Estes fatores justificam a criação de um guia que agrupa e classifica
            essas informações.
            A classificação facetada nos permite realizar determinados filtros
            conforme as características mapeadas.
            Desta forma, a partir de um determinado cenário, os usuários podem
            definir as características associadas ao seu contexto e obter as
            técnicas mais adequadas para sua configuração.
          </p>

          <h2 class="subtitle"><b-icon icon="target"></b-icon> Público alvo</h2>
          <div class="columns">
            <div class="column">
              <p>Pessoas em geral que procuram informações sobre as técnicas e
                como elas podem ser aplicadas no contexto de elicitação.</p>

              <figure class="image">
                
              </figure>

              <p>O guia também é útil para as pessoas que tem o objetivo de se
                informar sobre o assunto. Neste grupo estão inclusos
                estudantes, profissionais da área, entre outras pessoas.</p>
            </div>

```

```

<div class="column">
  <p>Profissionais da área que buscam técnicas para resolver
    situações enfrentadas durante a elicitação de requisito para
    um determinado contexto.</p>

  <figure class="image">
    
  </figure>

  <p>Com informações sobre o contexto em mãos, é possível utilizar a
    classificação para encontrar as técnicas mais adequadas e
    obter informações detalhadas sobre as mesmas.</p>
</div>
</div>

<h2 class="subtitle"><b-icon icon="target"></b-icon> Sobre a pesquisa</h2>
<p>Uma das etapas deste projeto foi a execução de uma revisão sistemática
  para coleta de informações sobre as técnicas e posterior classificação
  das mesmas através da técnica de facetas. Foram encontradas mais de
  90 técnicas referenciadas pelos autores como parte da etapa de
  elicitação de requisitos.</p>
<div class="columns">

  <div class="column">
    <figure class="image">
      
    </figure>

    Estas são as técnicas mais encontradas nos trabalhos coletados. <
      sup>[1]</sup>
  </div>
  <div class="column">
    <figure class="image">
      
    </figure>

    Esta nuvem de palavras contém todas técnicas encontradas e seu
    tamanho é proporcional ao número de artigos que às citam. <sup>
      >[1]</sup>
  </div>
</div>

<hr>

<div class="row content">
  <references title="Referências" :items="[<code> 1, description: '
    IGNACIO, Rafael C. Guia Facetado de Técnicas de Elicitação de
    Requisitos. Santa Catarina. 2018']"></references>
</div>
</section>
</div>
</section>
</section>
</template>

// resources/assets/js/pages/ClassificationPage.js

<script>
import { mapMutations } from 'vuex'
import VueMarkdown from 'vue-markdown'
import Card from '../components/Card'
import Breadcrumb from '../components/Breadcrumb'

export default {
  components: {
    VueMarkdown,
    Card,
    Breadcrumb
  }
}

```

```

    },
    data: function () {
      return {
        classification: {},
        entities: [],
        errors: [],
        breadcrumb: []
      };
    },
    created () {
      const slug = this.$route.params.classification
      const url = this.$store.getters.getUrl

      this.loading()

      Promise.all([
        axios.get(url + '/api/classifications/' + slug),
        axios.get(url + '/api/classifications/' + slug + '/entities')
      ]).then(([responseClassification, responseEntities]) => {
        this.classification = responseClassification.data
        this.entities = responseEntities.data

        this.loaded()
      }).catch((error) => this.errors = error.response.data.errors)
    },
    methods: {
      ...mapMutations([
        'loading',
        'loaded'
      ])
    }
  }
</script>

<template>
  <section v-if="this.$store.getters.isLoaded">
    <section class="hero">
      <div class="hero-body">
        <breadcrumb :items="[
          { url: '/app', title: 'Guia Facetado de Engenharia de Requisitos' },
          { url: '#', title: classification.title, active: true }
        ]"></breadcrumb>

        <div class="row content">
          <h1 class="title"><b-icon icon="bookmark"></b-icon> {{ this.classification
            .title }}</h1>
          <vue-markdown :source="classification.description"></vue-markdown>
        </div>

        <div class="row content">
          <h2 class="subtitle" v-if="entities.length > 0"><b-icon icon="adjust"></b-
            icon> Exemplos encontrados nesta seção do guia</h2>
          <h2 class="subtitle" v-else><b-icon icon="close"></b-icon> Nenhum exemplo
            encontrado para esta classificação</h2>
        </div>

        <div class="row content">
          <div class="columns is-multiline">
            <div class="column is-4 is-6-tablet is-12-mobile" v-for="entity in
              entities.slice(0, 4)">
              <card :title="entity.title"
                :content="entity.short_description"
                :action="'/app/classificacoes/' + $route.params.
                  classification + '/entidades/' + entity.slug"></card>
            </div>
          </div>
        </div>

        <div class="row content has-text-centered" v-if="entities.length > 3">
          <router-link :to="{ path: '/app/classificacoes/' + $route.params.
            classification + '/entidades' }" class="is-medium">
            <b-icon icon="plus" size="is-small"></b-icon> <span>Visualizar {{ this
              .classification.classification_type }}</span>
          </router-link>
        </div>
      </section>
    </section>
  </template>

```



```

        </router-link>
      </div>
    </div>
  </section>
</section>
</template>

// resources/assets/js/pages/EntitiesPage.js

<script>
import { mapMutations } from 'vuex'
import ModalForm from '../components/ModalForm'
import Card from '../components/Card'
import Breadcrumb from '../components/Breadcrumb'

export default {
  components: {
    ModalForm,
    Card,
    Breadcrumb
  },
  data: function () {
    return {
      isComponentModalActive: false,
      formProps: {},
      classification: {},
      entities: [],
      facetGroups: [],
      filteredEntities: [],
      errors: []
    }
  },
  created () {
    const slug = this.$route.params.classification;
    const url = this.$store.getters.getUrl

    this.loading()
    // localforage

    Promise.all([
      axios.get(url + '/api/classifications/' + slug),
      axios.get(url + '/api/classifications/' + slug + '/entities'),
      axios.get(url + '/api/facet_groups/' + slug)
    ]).then(([responseClassification, responseEntities, responseFacetGroups]) => {
      this.classification = responseClassification.data;
      this.entities = responseEntities.data;
      this.facetGroups = responseFacetGroups.data;

      this.filteredEntities = this.entities;
      this.loaded()
    }).catch((error) => this.errors = error.response.data.errors);
  },
  methods: {
    ...mapMutations([
      'resetFilters',
      'loading',
      'loaded'
    ]),
    filter: function () {
      const slug = this.$route.params.classification
      const url = this.$store.getters.getUrl

      this.loading()

      axios.post(url + '/api/classifications/' + slug + '/entities', this.$store.
        getters.getFilters, {
          'Content-Type': 'application/json'
        }).then((responseEntities) => {
          this.filteredEntities = responseEntities.data

          this.loaded()
        }).catch((error) => this.errors = error.response.data.errors)
    },
  },

```

```

        reset: function () {
            this.resetFilters()
            this.filteredEntities = this.entities
            this.filter()
        }
    };
</script>

<template>
    <section v-if="this.$store.getters.isLoaded">
        <section class="hero">
            <div class="hero-body">
                <breadcrumb :items="[
                    { url: '/app', title: 'Guia Facetado de Engenharia de Requisitos' },
                    { url: '/app/classificacoes/' + classification.slug, title: classification
                        .title },
                    { url: '#', title: 'Técnicas Mapeadas', active: true }
                ]"></breadcrumb>

                <div class="row content">
                    <h1 class="title">
                        <div class="field has-addons is-pulled-right">
                            <p class="control">
                                <button class="button is-danger is-medium" @click="reset()">
                                    <b-icon icon="eraser"></b-icon> <span>Limpar filtros</span>
                                </button>
                            </p>
                            <p class="control">
                                <button class="button is-primary is-medium" @click="
                                    isComponentModalActive = true">
                                    <b-icon icon="filter"></b-icon> <span>Filtrar</span>
                                </button>
                            </p>
                        </div>

                        <b-icon icon="bookmark"></b-icon> {{ classification.title }}
                    </h1>

                    <p class="has-text-justified">Nesta página são apresentados os elementos
                        que compõe a classificação acessada.</p>
                </div>

                <div class="row content" v-if="filteredEntities.length">
                    <div class="columns is-multiline">
                        <div class="column is-12">
                            <h2 class="subtitle"><b-icon icon="format-list-bulleted"></b-icon>
                                Registros encontrados</h2>
                        </div>
                        <div class="column is-4 is-6-tablet is-12-mobile" v-for="entity in
                            filteredEntities">
                            <card :title="entity.title"
                                :content="entity.short_description"
                                :action="'/app/classificacoes/' + $route.params.
                                    classification + '/entidades/' + entity.slug"></card>
                        </div>
                    </div>
                </div>

                <div class="row content" v-else>
                    <p v-if="$store.getters.getFiltersValues.length">Nenhum registro
                        encontrado para o filtro informado.</p>
                    <p v-else>Nenhum registro encontrado para esta classificação.</p>
                </div>
            </div>

            <b-modal :active.sync="isComponentModalActive" class="modal modal-full-screen
                modal-fx-fadeInScale" width="100%">
                <modal-form v-bind="formProps"
                    title="Selecione os filtros conforme as seguintes facetas"
                    :horizontal-data="facetGroups.filter( ( elem ) => elem.layout ===
                        'horizontal' )"

```

```

        :vertical-data="facetGroups.filter( ( elem ) => elem.layout === '
          vertical' )"
        @filter="filter"
        @reset="reset"></modal-form>
      </b-modal>
    </section>
  </section>
</template>

// resources/assets/js/pages/EntityPage.js

<script>
import { mapMutations } from 'vuex'
import VueMarkdown from 'vue-markdown'
import References from '../components/References'
import Classification from '../components/Classification'
import Breadcrumb from '../components/Breadcrumb'

export default {
  components: {
    VueMarkdown,
    Classification,
    References,
    Breadcrumb
  },
  data: function () {
    return {
      classification: {},
      entity: {},
      values: [],
      references: [],
      facets: [],
      errors: []
    }
  },
  created() {
    const classificationSlug = this.$route.params.classification
    const entitySlug = this.$route.params.entity
    const url = this.$store.getters.getUrl

    this.loading()

    Promise.all([
      axios.get(url + '/api/classifications/' + classificationSlug),
      axios.get(url + '/api/entities/' + entitySlug),
      axios.get(url + '/api/entities/' + entitySlug + '/values'),
      axios.get(url + '/api/entities/' + entitySlug + '/references'),
      axios.get(url + '/api/facet_groups/' + classificationSlug)
    ]).then(([responseClassification, responseEntities, responseEntityValues,
      responseEntityReferences, responseFacets]) => {
      this.classification = responseClassification.data
      this.entity = responseEntities.data
      this.values = responseEntityValues.data
      this.references = responseEntityReferences.data
      this.facets = responseFacets.data

      axios.get(url + '/api/entities/page_views/' + this.entity.id)
        .catch((error) => this.errors = error.response.data.errors)

      this.loaded()
    }).catch((error) => this.errors = error.response.data.errors)
  },
  methods: {
    ...mapMutations([
      'loading',
      'loaded'
    ])
  }
};
</script>

<template>
  <section v-if="this.$store.getters.isLoaded">

```

```

<section class="hero">
  <div class="hero-body">
    <breadcrumb :items="[
      { url: '/app', title: 'Guia Facetado de Engenharia de Requisitos' },
      { url: '/app/classificacoes/' + classification.slug, title:
        classification.title },
      { url: '/app/classificacoes/' + classification.slug + '/entidades',
        title: 'Técnicas Mapeadas' },
      { url: '#', title: entity.title, active: true }
    ]"></breadcrumb>

    <div class="row content">
      <h1 class="title"><b-icon icon="bookmark-outline"></b-icon> {{ entity.
        title }}</h1>

      <div class="text-2-columns has-text-justified">
        <vue-markdown :source="entity.description"></vue-markdown>

        <template v-if="entity.images_array && entity.images_array.length">
          <h2>Figuras</h2>
          <div v-for="image in entity.images_array">
            <lightbox
              style="width: 40em"
              :thumbnail="image.src"
              :images="[image.src]"
            >
              <lightbox-default-loader slot="loader"/>
            </lightbox>
            <small>{{ image.title }}</small>
            <br><br>
          </div>
        </template>
      </div>
    </div>

    <hr>

    <div class="row content">
      <div class="columns is-multiline">
        <div class="column is-12 is-6-desktop">
          <h2 class="subtitle"><b-icon icon="plus" class="has-text-success"
            ></b-icon> <span>Prós</span></h2>

          <vue-markdown :source="entity.pros || 'Informação não encontrada
            :,'"></vue-markdown>
        </div>
        <div class="column is-12 is-6-desktop">
          <h2 class="subtitle"><b-icon icon="minus" class="has-text-danger"
            ></b-icon> <span>Contras</span></h2>

          <vue-markdown :source="entity.cons || 'Informação não encontrada
            :,'"></vue-markdown>
        </div>
      </div>
    </div>

    <hr>

    <div class="row">
      <classification :values="values" :items="facets"></classification>
    </div>

    <hr>

    <div class="row content">
      <references title="Referências" :items="references"></references>
    </div>
  </div>
</section>
</section>
</template>

// resources/assets/js/pages/FacetsPage.js

```

```

<script>
  import { mapMutations } from 'vuex'
  import References from '../components/References'
  import Breadcrumb from '../components/Breadcrumb'

  export default {
    components: {
      References,
      Breadcrumb
    },
    data: function () {
      return {
        classification: {},
        facets: [],
        references: [],
        errors: []
      }
    },
    created() {
      const slug = this.$route.params.classification
      const url = this.$store.getters.getUrl

      this.loading()

      Promise.all([
        axios.get(url + '/api/classifications/' + slug),
        axios.get(url + '/api/classifications/' + slug + '/facets'),
        axios.get(url + '/api/classifications/' + slug + '/facets/references')
      ]).then([responseClassification, responseFacets, responseFacetsReferences] => {
        this.classification = responseClassification.data
        this.facets = responseFacets.data
        this.references = responseFacetsReferences.data

        this.loaded()
      }).catch((error) => this.errors = error.response.data.errors)
    },
    methods: {
      ...mapMutations([
        'loading',
        'loaded'
      ])
    }
  };
</script>

<template>
  <section v-if="this.$store.getters.isLoaded">
    <section class="hero">
      <div class="hero-body">
        <breadcrumb :items="[
          { url: '/app', title: 'Guia Facetado de Engenharia de Requisitos' },
          { url: '/app/classificacoes/' + classification.slug, title: classification.title },
          { url: '#', title: 'Facetas de Classificação', active: true }
        ]"></breadcrumb>

        <section class="row content">
          <h1 class="title"><b-icon icon="bookmark"></b-icon> Facetas de classificação</h1>
          <p class="has-text-justified">Nesta página são apresentadas as facetas que determinam a classificação desta seção.</p>
        </section>

        <section class="row">
          <b-table
            :data="this.facets"
            paginated
            per-page="10"
            detailed
            detail-key="id">

```

```

<template slot-scope="props">
  <b-table-column field="id" label="#" width="40" numeric>
    {{ props.row.id }}
  </b-table-column>

  <b-table-column field="user.title" label="Título">
    {{ props.row.title }}
  </b-table-column>
</template>

<template slot="detail" slot-scope="props">
  <article class="media">
    <div class="media-content">
      <div class="content">
        <span>
          <small>Descrição</small>
          <br>
          <span>{{ props.row.description }}</span>
          <br><br>
          <small>Opções:</small>
          <ul v-if="props.row.values.length">
            <li v-for="value in props.row.values">
              <div><b-icon icon="chevron-right" size="is-small"></b-icon> <span>{{ value.title }}</span></div>
              <div v-if="value.description" class="inner-list">
                <small>{{ value.description }}</small>
              </div>
            </li>
          </ul>
        </span>
      </div>
    </div>
  </article>
</template>
</b-table>
</section>

<section class="row content">
  <references title="Referências" :items="references"></references>
</section>
</div>
</section>
</section>
</template>

<style scoped>
  ul { list-style: none; }
  .inner-list { margin-left: 12px;padding-left: 13px;border-left: 5px #CCC solid; }
</style>

// resources/assets/js/app.js

require('./bootstrap');
window.Vue = require('vue');

import Vue from 'vue'
import Vuex from 'vuex'
import VueRouter from 'vue-router'
import VueMarkdown from 'vue-markdown'
import VueLoading from 'vue-loading-overlay'
import Sidebar from './components/Sidebar'
import Buefy from 'buefy'
import Lightbox from 'vue-pure-lightbox'
import { APP_URL } from './utils/config'

import { routes } from './utils/routes';

Vue.use(Vuex);
Vue.use(VueRouter);
Vue.use(Buefy);

```

```

Vue.use(VueMarkdown);
Vue.use(VueLoading);
Vue.use(Lightbox);

const router = new VueRouter({
  mode: 'history',
  routes
});

const store = new Vuex.Store({
  state: {
    filters: {
      mode: 'restrict',
      values: []
    },
    defaultValues: [],
    isLoading: false
  },
  mutations: {
    updateMode (state, mode) {
      state.filters.mode = mode
    },
    addFilter (state, filter) {
      state.filters.values.forEach((item, index) => {
        if (item.name === filter.name) {
          state.filters.values.splice(index, 1)
        }
      })

      state.filters.values.push(filter)
    },
    addDefaultValue (state, filter) {
      state.defaultValues.forEach((item, index) => {
        if (item.name === filter.name) {
          state.defaultValues.splice(index, 1)
        }
      })

      state.defaultValues.push(filter)
    },
    resetFilters (state) {
      state.filters = {
        mode: 'restrict',
        values: []
      }
    },
    loading (state) {
      state.isLoading = true
    },
    loaded (state) {
      state.isLoading = false
    }
  },
  getters: {
    getFilterValueByName: (state) => (name) => {
      let filter = state.filters.values.filter((item) => {
        return item.name === name
      })

      if (filter.length) {
        return filter.shift().value
      }

      filter = state.defaultValues.filter((item) => {
        return item.name === name
      })

      return filter.length ? filter.shift().value : null
    },
    getFilters: (state) => {
      return state.filters || []
    },
    getFiltersValues: (state) => {

```

```

        return state.filters.values || []
      },
      getMode: (state) => {
        return state.filters.mode
      },
      isLoading: (state) => {
        return state.isLoading
      },
      isLoading: (state) => {
        return ! state.isLoading
      },
      getUrl: () => {
        return APP_URL
      }
    }
  });

const app = new Vue({
  el: '#app',
  router,
  store,
  components: {
    Sidebar,
    Loading: VueLoading
  },
  computed: {
    url () {
      return this.$store.getters.getUrl
    }
  },
  data() {
    return {}
  }
});

// resources/views/classification/edit.blade.php

@extends('layouts.general')

@section('breadcrumb')
  <nav class="breadcrumb" aria-label="breadcrumbs">
    <ul>
      <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
      <li><a href="{{ route('classifications') }}" aria-current="page">{{ __('Classificações') }}</a></li>
      <li class="is-active"><a href="#" aria-current="page">{{ (int) $id > 0 ? __('Editar classificação') : __('Nova classificação') }}</a></li>
    </ul>
  </nav>
@endsection

@section('content')
  <div class="columns has-background-white">
    <div class="column is-8">
      <form id="form" action="{{ (int) $id > 0 ? route('classifications.update', ['id' => (int) $id]) : route('classifications.store') }}" method="POST">
        @if ((int) $id > 0)
          @method('PUT')
        @endif

        @csrf

        <div class="field">
          <label class="label">{{ __('Titulo') }}</label>
          <div class="control">
            <input name="title" class="input is-medium" type="text" placeholder="{{ __('Informe o title') }}" value="{{ $classification ? $classification->title : null }}">
          </div>
        </div>

        <div class="field">

```



```

<label class="label">{{ __('Slug') }}</label>
<div class="control">
    <input name="slug" class="input is-medium" type="text" placeholder="{{
        __('Informe o slug') }}" value="{{ $classification ?
            $classification->slug : null }}">
    </div>
</div>

<div class="field">
    <label class="label">{{ __('Descrição') }}</label>
    <div class="control">
        <textarea name="description" class="textarea" placeholder="Textarea"
            >{{ $classification ? $classification->description : null }}</
            textarea>
    </div>
</div>

<div class="field">
    <label class="label">{{ __('Tipo de classificação') }}</label>
    <div class="control">
        <input name="classification_type" class="input is-medium" type="text"
            placeholder="{{ __('Informe o tipo da classificação') }}" value="{{
                $classification ? $classification->classification_type : null
            }}">
    </div>
</div>

<div class="field">
    <label class="label">{{ __('Título no menu') }}</label>
    <div class="control">
        <input name="main_menu" class="input is-medium" type="text"
            placeholder="{{ __('Informe o título do menu') }}" value="{{
                $classification ? $classification->main_menu : null }}">
    </div>
</div>

<div class="field is-grouped">
    <div class="control">
        <button class="button is-link">{{ (int) $id > 0 ? __('Alterar') : __('
            Cadastrar ') }}</button>
    </div>
    <div class="control">
        <a href="{{ route('classifications') }}" class="button is-text">{{ __('
            Cancelar') }}</a>
    </div>
</div>
</form>
</div>
</div>
@endsection

// resources/views/classification/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></
                li>
            <li class="is-active"><a href="#" aria-current="page">{{ __('Classificações') }}</
                a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <h2 class="subtitle">
        <a href="{{ route('classifications.edit', ['id' => 0]) }}" class="button is-medium is-
            pulled-right">
            <span class="icon"><span class="mdi mdi-check"></span></span> <span>{{ __('Nova
                classificação') }}</span>
        </a>

```

```

        {{ __('Classificações') }}
    </h2>
    <section class="info-tiles">
        @forelse ($classifications as $classification)
            <div class="card">
                <header class="card-header">
                    <p class="card-header-title">{{ $classification->title }}</p>
                </header>
                <div class="card-content">
                    <div class="content">
                        {{ $classification->description }}
                    </div>
                </div>
                <footer class="card-footer">
                    <a href="{{ route('classifications.edit', ['id' => $classification->id]) }}" class="card-footer-item">
                        <span class="icon"><span class="mdi mdi-pencil"></span></span> {{ __('Editar') }}
                    </a>
                    <a href="{{ route('classifications.entities', ['classificationId' => $classification->id]) }}" class="card-footer-item">
                        <span class="icon"><span class="mdi mdi-application"></span></span> <span class="badge is-badge-info" data-badge="{{ count($classification->entities) }}">{{ __('Técnicas') }}</span>
                    </a>
                    <a href="{{ route('classifications.facets', ['classificationId' => $classification->id]) }}" class="card-footer-item">
                        <span class="icon"><span class="mdi mdi-adjust"></span></span> <span class="badge is-badge-info" data-badge="{{ count($classification->facets) }}">{{ __('Facetas') }}</span>
                    </a>
                    @if ($classification->published)
                        <a href="#" @click="unpublishClassification({{ $classification->id }}" class="card-footer-item">
                            <span class="icon" v-if="isLoading()"><span class="mdi mdi-spin mdi-loading"></span></span>
                            <span class="icon" v-if="! isLoading()"><span class="mdi mdi-arrow-expand-down"></span></span> {{ __('Despublicar') }}
                        </a>
                    @else
                        <a href="#" @click="publishClassification({{ $classification->id }}" class="card-footer-item">
                            <span class="icon" v-if="isLoading()"><span class="mdi mdi-spin mdi-loading"></span></span>
                            <span class="icon" v-if="! isLoading()"><span class="mdi mdi-arrow-expand-up is-success"></span></span> {{ __('Publicar') }}
                        </a>
                    @endif
                </footer>
            </div>
        @empty
            <div class="notification is-info">{{ __('Nenhuma classificação encontrada.') }}</div>
        @endforelse
    </section>
@endsection

// resources/views/entity/edit.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li><a href="{{ route('classifications') }}" aria-current="page">{{ __('Classificações') }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ $classification->title }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ (int) $id > 0 ? __('Editar técnica') : __('Nova técnica') }}</a></li>
        </ul>
    </nav>

```

```

        </ul>
    </nav>
@endsection

@section('content')
    <div class="columns has-background-white">
        <div class="column is-12">
            <form id="form" action="{{ (int) $id > 0 ?
                route('entities.update', ['classificationId' => (int) $classificationId, 'id'
                    => (int) $id]) :
                route('entities.store', ['classificationId' => (int) $classificationId]) }}"
                method="POST">
                @if ((int) $id > 0)
                    @method('PUT')
                @endif
                @csrf
                <input type="hidden" name="classification_id" value="{{ $classificationId }}">
                @if ($entity) <input type="hidden" name="id" value="{{ $id }}"> @endif

            <div class="tabs is-medium">
                <ul>
                    <li class="is-active" @click="tab='default'">
                        <a href="#"><span>Definição</span></a>
                    </li>
                    @if ($entity)
                    <li>
                        <a href="{{ route('entities.classification', [
                            'classificationId' => $classificationId,
                            'id' => $id,
                        ]) }}"><span>Classificação</span></a>
                    </li>
                    <li>
                        <a href="{{ route('entities.references', [
                            'classificationId' => $classificationId,
                            'id' => $id,
                        ]) }}"><span>Referências</span></a>
                    </li>
                    @endif
                </ul>
            </div>

            <div class="box help-content has-background-white">
                <div class="field">
                    <label class="label">{{ __('Título') }}</label>
                    <div class="control">
                        <input name="title" class="input is-medium" type="text"
                            placeholder="{{ __('Informe o título') }}" value="{{ $entity ?
                                $entity->title : null }}">
                    </div>
                </div>

                <div class="field">
                    <label class="label">{{ __('Slug') }}</label>
                    <div class="control">
                        <input name="slug" class="input is-medium" type="text" placeholder
                            = "{{ __('Informe o slug') }}" value="{{ $entity ? $entity->
                                slug : null }}">
                    </div>
                </div>

                <div class="field">
                    <label class="label">{{ __('Resumo') }}</label>
                    <div class="control">
                        <textarea name="short_description" rows="2" class="textarea"
                            placeholder="{{ __('Informe a descrição') }}">{{ $entity ?
                                $entity->short_description : null }}</textarea>
                    </div>
                </div>

                <div class="field">
                    <label class="label">{{ __('Descrição') }}</label>
                    <div class="control">
                        <textarea name="description" class="textarea" placeholder="{{ __('

```

```

        Informe a descrição') }}">{{ $entity ? $entity->description :
        null }}</textarea>
    </div>
</div>

<div class="field">
    <label class="label">{{ __( 'Prós' ) }}</label>
    <div class="control">
        <textarea name="pros" class="textarea" placeholder="{{ __( 'Informe
        os pontos positivos' ) }}">{{ $entity ? $entity->pros : null
        }}</textarea>
    </div>
</div>

<div class="field">
    <label class="label">{{ __( 'Contras' ) }}</label>
    <div class="control">
        <textarea name="cons" class="textarea" placeholder="{{ __( 'Informe
        os pontos negativos' ) }}">{{ $entity ? $entity->cons : null
        }}</textarea>
    </div>
</div>

<div class="field">
    <label class="label">{{ __( 'Imagens' ) }}</label>
    <div class="control">
        <textarea name="images" class="textarea" placeholder="{{ __( '
        Informe as imagens' ) }}">{{ $entity ? $entity->images : null
        }}</textarea>
    </div>
</div>
</div>

<div class="field is-grouped">
    <div class="control">
        <button class="button is-link">{{ (int) $id > 0 ? __( 'Alterar' ) : __( '
        Cadastrar' ) }}</button>
    </div>
    <div class="control">
        <button class="button is-text"></button>
        <a href="{{ route('classifications.entities', ['classificationId' =>
        $classificationId] ) }}" class="button is-text">{{ __( 'Cancelar' )
        }}</a>
    </div>
</div>
</form>
</div>
</div>
@endsection

// resources/views/entity/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home' ) }}" aria-current="page">{{ __( 'Dashboard' ) }}</a></li>
            <li><a href="{{ route('classifications' ) }}" aria-current="page">{{ __( '
            Classificações' ) }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ $classification->title
            }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ $classification->
            classification_type }}</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <h2 class="subtitle">
        <a href="{{ route('entities.edit', [
            'classificationId' => $classification->id,

```

```

        'id' => 0
    ]) }}" class="button is-medium is-pulled-right">
        <span class="icon"><span class="mdi mdi-check"></span></span> <span>{{ __( 'Nova
            Técnica' ) }}</span>
    </a>

    {{ $classification->classification_type }}
</h2>
<table class="table is-fullwidth is-striped is-hoverable is-bordered">
    <thead>
        <tr>
            <th style="width: 5%;"><abbr title="{{ __( 'Código' ) }}">#</abbr></th>
            <th style="width: 20%;">{{ __( 'Título' ) }}</th>
            <th>{{ __( 'Descrição' ) }}</th>
            <th class="has-text-centered" style="width: 9%;">{{ __( 'Ações' ) }}</th>
        </tr>
    </thead>
    <tbody>
        @foreach ( $entities as $entity )
            <tr>
                <td>{{ $entity->id }}</td>
                <td>{{ $entity->title }}</td>
                <td>{{ $entity->short_description }}</td>
                <td class="has-text-centered">
                    <a href="{{ route('entities.edit', [
                        'classificationId' => $entity->classification_id,
                        'id' => $entity->id
                    ]) }}" class="button is-small">
                        <span class="icon"><span class="mdi mdi-pencil"></span></span></a>
                    @if ( $entity->published )
                        <a href="#" @click="unpublishEntity({{ $entity->classification_id }},
                            {{ $entity->id }})" class="button is-small">
                            <span class="icon" v-if="isLoading()"><span class="mdi mdi-spin
                                mdi-loading"></span></span>
                            <span class="icon" v-if="! isLoading()"><span class="mdi mdi-arrow
                                -expand-down"></span></span>
                        </a>
                    @else
                        <a href="#" @click="publishEntity({{ $entity->classification_id }}, {{
                            $entity->id }})" class="button is-small">
                            <span class="icon" v-if="isLoading()"><span class="mdi mdi-spin
                                mdi-loading"></span></span>
                            <span class="icon" v-if="! isLoading()"><span class="mdi mdi-arrow
                                -expand-up is-success"></span></span>
                        </a>
                    @endif
                </td>
            </tr>
        @endforeach
    </tbody>
    <tfoot>
        <tr>
            <td colspan="4">
                {{ $entities->links('vendor.pagination.simple-default') }}
            </td>
        </tr>
    </tfoot>
</table>
@endsection

// resources/views/entity/classification/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __( 'Dashboard' ) }}</a></li>
            <li><a href="{{ route('classifications') }}" aria-current="page">{{ __( '
                Classificações' ) }}</a></li>
            <li><a href="{{ route('entities.edit', ['classificationId' => $classificationId, '

```

```

        id' => $entityId]) }}" aria-current="page">{{ $entity->title }}</a></li>
    <li class="is-active"><a href="#" aria-current="page">Referências</a></li>
</ul>
</nav>
@endsection

@section('content')
    <div class="columns has-background-white">
        <div class="column is-12">
            <div class="tabs is-medium">
                <ul>
                    <li>
                        <a href="{{ route('entities.edit', [
                            'classificationId' => $classificationId,
                            'id' => $entityId,
                        ]) }}"><span>Definição</span></a>
                    </li>
                    <li class="is-active">
                        <a href="#"><span>Classificação</span></a>
                    </li>
                    <li>
                        <a href="{{ route('entities.references', [
                            'classificationId' => $classificationId,
                            'id' => $entityId,
                        ]) }}"><span>Referências</span></a>
                    </li>
                </ul>
            </div>

            <form id="form" action="{{ route('entities.classification', ['classificationId' =>
                $classificationId, 'entityId' => (int) $entityId]) }}" method="POST">
                @csrf
                <input type="hidden" name="classification_id" value="{{ $classificationId }}">
                <input type="hidden" name="entity_id" value="{{ $entityId }}">

                <div class="has-background-white">
                    @foreach($facetsGroups as $facetGroup)
                        <section class="box">
                            <p class="subtitle">{{ $facetGroup->title }}</p>
                            <hr>

                            <div class="columns is-multiline">
                                @foreach($facetGroup->facets as $facet)
                                    <div class="column is-6">
                                        <section>
                                            <b>{{ $facet->title }}</b> <hr>

                                            @foreach($facet->values as $value)
                                                <div class="field">
                                                    <div class="control">
                                                        <label class="checkbox">
                                                            <input type="checkbox" name="
                                                                values[]" value="{{ $value->id
                                                                }}">
                                                            @if (in_array($value->id,
                                                                $entityValues)) checked
                                                            @endif <{{ $value->title
                                                                }}>
                                                        </label>
                                                    </div>
                                                </div>
                                            @endforeach

                                        </section>
                                    </div>
                                <div class="is-clearfix"></div>
                            @endforeach
                        </div>
                    </section>
                </div>

                <div class="is-clearfix"></div>
            </div>
        </div>

        <div id="form" action="{{ route('entities.classification', ['classificationId' =>
            $classificationId, 'entityId' => (int) $entityId]) }}" method="POST">
            @csrf
            <input type="hidden" name="classification_id" value="{{ $classificationId }}">
            <input type="hidden" name="entity_id" value="{{ $entityId }}">

            <div class="has-background-white">
                @foreach($facetsGroups as $facetGroup)
                    <section class="box">
                        <p class="subtitle">{{ $facetGroup->title }}</p>
                        <hr>

                        <div class="columns is-multiline">
                            @foreach($facetGroup->facets as $facet)
                                <div class="column is-6">
                                    <section>
                                        <b>{{ $facet->title }}</b> <hr>

                                        @foreach($facet->values as $value)
                                            <div class="field">
                                                <div class="control">
                                                    <label class="checkbox">
                                                        <input type="checkbox" name="
                                                            values[]" value="{{ $value->id
                                                            }}">
                                                        @if (in_array($value->id,
                                                            $entityValues)) checked
                                                        @endif <{{ $value->title
                                                            }}>
                                                    </label>
                                                </div>
                                            </div>
                                        @endforeach

                                    </section>
                                </div>
                            <div class="is-clearfix"></div>
                        @endforeach
                    </div>
                </section>
            </div>

            <div class="is-clearfix"></div>
        </div>
    </div>
</div>

```

```

        <div class="field is-grouped">
            <div class="control">
                <button class="button is-link">{{ __('Alterar') }}</button>
            </div>
            <div class="control">
                <button class="button is-text"></button>
                <a href="{{ route('entities.edit', ['classificationId' =>
                    $classificationId, 'id' => $entityId]) }}" class="button is-
                    text">{{ __('Cancelar') }}</a>
            </div>
        </div>
    </div>
</form>
</div>
</div>
@endsection

// resources/views/entity/references/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li><a href="{{ route('classifications') }}" aria-current="page">{{ __('Classificações') }}</a></li>
            <li><a href="{{ route('entities.edit', ['classificationId' => $classificationId, 'id' => $entityId]) }}" aria-current="page">{{ $entity->title }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">Referências</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <div class="columns has-background-white">
        <div class="column is-12">
            <div class="tabs is-medium">
                <ul>
                    <li>
                        <a href="{{ route('entities.edit', [
                            'classificationId' => $classificationId,
                            'id' => $entityId,
                        ]) }}"><span>Definição</span></a>
                    </li>
                    <li>
                        <a href="{{ route('entities.classification', [
                            'classificationId' => $classificationId,
                            'id' => $entityId,
                        ]) }}"><span>Classificação</span></a>
                    </li>
                    <li class="is-active">
                        <a href="#"><span>Referências</span></a>
                    </li>
                </ul>
            </div>

            <form id="form" action="{{ route('entities.references', ['classificationId' =>
                $classificationId, 'entityId' => (int) $entityId]) }}" method="POST">
                @csrf
                <input type="hidden" name="classification_id" value="{{ $classificationId }}">
                <input type="hidden" name="entity_id" value="{{ $entityId }}">

                <div class="help-content has-background-white">
                    <div class="columns">
                        <div class="column is-2">
                            <div class="control">
                                <label class="label">{{ __('Código') }}</label>
                                <input name="code" class="input" type="text" placeholder="Informe a código da referência">
                            </div>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>

```

```

<div class="column">
  <div class="field">
    <div class="control is-expanded">
      <label class="label">{{ __('Referência') }}</label>
      <textarea name="description" rows="1" class="textarea"
        placeholder="{{ __('Informe a descrição da referência') }}"></textarea>
    </div>
  </div>
</div>
<div class="column is-1">
  <div class="field">
    <div class="control">
      <label class="label">&nbsp;</label>
      <button class="button is-link">{{ __('Cadastrar') }}</button>
    </div>
  </div>
</div>
</div>
</div>
</form>

<hr>

<table class="table is-fullwidth is-striped is-hoverable is-bordered">
  <thead>
    <tr>
      <th style="width: 5%;"><abbr title="{{ __('Código') }}">#</abbr></th>
      <th>{{ __('Descrição') }}</th>
      <th class="has-text-centered" style="width: 9%;">{{ __('Ações') }}</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($entity->references as $reference)
      <tr>
        <td>{{ $reference->pivot->code }}</td>
        <td>{{ $reference->description }}</td>
        <td class="has-text-centered">
          <a class="button is-small delete-button"
            data-target="modal" aria-haspopup="true" data-id="{{ $reference->id }}"
            data-classification="{{ $entityId }}"
            onclick="event.preventDefault();">
            <span class="icon"><span class="mdi mdi-delete"></span></span>
          </a>
        </td>
      </tr>
    @endforeach
  </tbody>
  <tfoot>
    <tr>
      <td colspan="5">
        @if (count($entity->references) <= 0) Nenhuma referência associada a esta técnica @endif
        &nbsp;
      </td>
    </tr>
  </tfoot>
</table>
</div>
</div>

<div class="modal" id="modal">
  <div class="modal-background"></div>
  <div class="modal-card">
    <header class="modal-card-head">
      <p class="modal-card-title">{{ __('Remover valor') }}</p>
      <button class="delete modal-cancel-x" aria-label="close"></button>
    </header>
    <section class="modal-card-body">
      {{ __('Você tem certeza?') }}
    </section>
    <footer class="modal-card-foot">

```



```

        <button class="button is-danger modal-confirm" data-id="" data-classification=
            "" @click="attemptDetachEntityReference()">{{ __('Remover') }}</button>
        <button class="button modal-cancel">Cancelar</button>
    </footer>
</div>
</div>
</div>
@endsection

// resources/views/facet/edit.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li><a href="{{ route('classifications') }}" aria-current="page">{{ __('Classificações') }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ $classification->title }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ (int) $id > 0 ? __('Editar faceta') : __('Nova faceta') }}</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <div class="columns has-background-white">
        <div class="column is-12">
            <form id="form" action="{{ (int) $id > 0 ?
                route('facets.update', ['classificationId' => $classificationId, 'id' => (int) $id]) :
                route('facets.store', ['classificationId' => $classificationId]) }}" method="POST">
                @if ((int) $id > 0)
                    @method('PUT')
                @endif @csrf
                <input type="hidden" name="classification_id" value="{{ $classificationId }}">
                @if ($facet) <input type="hidden" name="id" value="{{ $id }}"> @endif

            <div class="tabs is-medium">
                <ul>
                    <li class="is-active">
                        <a href="#"><span>Definição</span></a>
                    </li>
                    @if ($facet)
                        <li>
                            <a href="{{ route('facets.values', [
                                'classificationId' => $classificationId,
                                'id' => $id,
                            ]) }}"><span>Valores possíveis</span></a>
                        </li>
                        <li>
                            <a href="{{ route('facets.references', [
                                'classificationId' => $classificationId,
                                'id' => $id,
                            ]) }}"><span>Referências</span></a>
                        </li>
                    @endif
                </ul>
            </div>

            <div class="box help-content has-background-white">
                <div v-if="facetastab === 'default'">
                    <div class="field">
                        <label class="label">{{ __('Título') }}</label>
                        <div class="control">
                            <input name="title" class="input is-medium" type="text"
                                placeholder="{{ __('Informe o título') }}" value="{{ $facet ? $facet->title : null }}">
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="field">
  <label class="label">{{ __('Slug') }}</label>
  <div class="control">
    <input name="slug" class="input is-medium" type="text"
      placeholder="{{ __('Informe o slug') }}" value="{{ $facet
        ? $facet->slug : null }}">
    </div>
  </div>

<div class="field">
  <label class="label">{{ __('Descrição') }}</label>
  <div class="control">
    <textarea name="description" rows="2" class="textarea"
      placeholder="{{ __('Informe a descrição') }}">{{ $facet ?
        $facet->description : null }}</textarea>
    </div>
  </div>

<div class="field">
  <label class="label">{{ __('Grupo') }} <a href="{{ route('
    facets_groups') }}" class="button is-small"><span class="icon"
    ><span class="mdi mdi-pencil"></span></span> <span>Gerenciar</
    span></a></label>
  <div class="control">
    <div class="select is-info">
      <select name="facet_group_id">
        <option>{{ __('Selecione um grupo') }}</option>
        @foreach($facet_groups as $groupId => $group)
          <option value="{{ $groupId }}" @if ($facet &&
            $groupId === $facet->facet_group_id) selected
            @endif>{{ $group }}</option>
        @endforeach
      </select>
    </div>
  </div>
</div>

<div class="field">
  <label class="label">{{ __('Tipo') }}</label>
  <div class="control">
    <div class="select is-info">
      <select name="type">
        <option>Selecione um tipo</option>
        <option @if ($facet && $facet->type === 'select')
          selected @endif value="select">Lista</option>
        <option @if ($facet && $facet->type === 'switch')
          selected @endif value="switch">Switch</option>
        <option @if ($facet && $facet->type === 'slider')
          selected @endif value="slider">Slider</option>
        <option @if ($facet && $facet->type === 'checkbox')
          selected @endif value="checkbox">Checkbox</option>
        <option @if ($facet && $facet->type === 'checkboxbutton')
          selected @endif value="checkboxbutton">Checkboxbutton</
          option>
        </select>
      </div>
    </div>
  </div>
</div>

<div class="field is-grouped">
  <div class="control">
    <button class="button is-link">{{ (int) $id > 0 ? __('Alterar') : __('
      Cadastrar') }}</button>
  </div>
  <div class="control">
    <a href="{{ route('classifications.facets', ['classificationId' =>
      $classificationId]) }}" class="button is-text">{{ __('Cancelar')
      }}</a>
  </div>
</div>

```

```

        </form>
    </div>
</div>
@endsection

// resources/views/facet/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li><a href="{{ route('classifications') }}" aria-current="page">{{ __('Classificações') }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ $classification->title }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">Facetas de Classificação</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <h2 class="subtitle">
        <a href="{{ route('facets.edit', [
            'classificationId' => $classification->id,
            'id' => 0
        ]) }}" class="button is-medium is-pulled-right">
            <span class="icon"><span class="mdi mdi-check"></span></span> <span>{{ __('Nova Faceta') }}</span>
        </a>

        Facetas de classificação
    </h2>
    <table class="table is-fullwidth is-striped is-hoverable is-bordered">
        <thead>
            <tr>
                <th style="width: 5%;"><abbr title="{{ __('Código') }}">#</abbr></th>
                <th style="width: 20%;">{{ __('Título') }}</th>
                <th>{{ __('Descrição') }}</th>
                <th class="has-text-centered" style="width: 9%;">{{ __('Ações') }}</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($facets as $facet)
                <tr>
                    <td>{{ $facet->id }}</td>
                    <td>{{ $facet->title }}</td>
                    <td>{{ $facet->description }}</td>
                    <td class="has-text-centered">
                        <a href="{{ route('facets.edit', [
                            'classificationId' => $facet->classification_id,
                            'id' => $facet->id
                        ]) }}" class="button is-small">
                            <span class="icon"><span class="mdi mdi-pencil"></span></span></span>
                        <a class="button is-small delete-button"
                            data-target="modal" aria-haspopup="true" data-id="{{ $facet->id }}"
                            data-classification="{{ $facet->classification_id }}"
                            onclick="event.preventDefault();">
                            <span class="icon"><span class="mdi mdi-delete"></span></span>
                        </a>
                    </td>
                </tr>
            @endforeach
        </tbody>
        <tfoot>
            <tr>
                <td colspan="4">
                    {{ $facets->links('vendor.pagination.simple-default') }}
                </td>
            </tr>
        </tfoot>
    </table>

```



```

        <label class="label">{{ __('Código') }}</label>
        <input name="code" class="input" type="text" placeholder="
            Informe a código da referência">
    </div>
</div>
<div class="column">
    <div class="field">
        <div class="control is-expanded">
            <label class="label">{{ __('Referência') }}</label>
            <textarea name="description" rows="1" class="textarea"
                placeholder="{{ __('Informe a descrição da referência
                    ') }}"></textarea>
        </div>
    </div>
</div>
<div class="column is-1">
    <div class="field">
        <div class="control">
            <label class="label">&nbsp;</label>
            <button class="button is-link">{{ __('Cadastrar') }}</
                button>
        </div>
    </div>
</div>
</div>
</div>
</form>

<hr>

<table class="table is-fullwidth is-striped is-hoverable is-bordered">
    <thead>
        <tr>
            <th style="width: 5%;"><abbr title="{{ __('Código') }}">#</abbr></th>
            <th>{{ __('Descrição') }}</th>
            <th class="has-text-centered" style="width: 9%;">{{ __('Ações') }}</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($facet->references as $reference)
            <tr>
                <td>{{ $reference->pivot->code }}</td>
                <td>{{ $reference->description }}</td>
                <td class="has-text-centered">
                    <a class="button is-small delete-button"
                        data-target="modal" aria-haspopup="true" data-id="{{ $reference
                            ->id }}" data-classification="{{ $facetId }}"
                        onclick="event.preventDefault();">
                        <span class="icon"><span class="mdi mdi-delete"></span></span>
                    </a>
                </td>
            </tr>
        @endforeach
    </tbody>
    <tfoot>
        <tr>
            <td colspan="5">
                @if (count($facet->references) <= 0) Nenhuma referência associada a
                    esta faceta @endif
                &nbsp;   
            </td>
        </tr>
    </tfoot>
</table>
</div>
</div>

<div class="modal" id="modal">
    <div class="modal-background"></div>
    <div class="modal-card">
        <header class="modal-card-head">
            <p class="modal-card-title">{{ __('Remover valor') }}</p>
            <button class="delete modal-cancel-x" aria-label="close"></button>

```



```

        </div>
    </div>
    <div class="field">
        <label class="label">{{ __( 'Slug' ) }}</label>
        <div class="control">
            <input name="slug" class="input is-medium" type="text" placeholder
                ="{{ __( 'Informe o slug' ) }}" value="{{ $value ? $value->slug
                    : null }}">
        </div>
    </div>
    <div class="field">
        <label class="label">{{ __( 'Valor' ) }}</label>
        <div class="control">
            <input name="value" class="input is-medium" type="text"
                placeholder="{{ __( 'Informe o valor correspondente a esta
                    característica' ) }}" value="{{ $value ? $value->value : null
                    }}">
        </div>
    </div>
    <div class="field">
        <label class="label">{{ __( 'Descrição' ) }}</label>
        <div class="control">
            <textarea name="description" rows="2" class="textarea" placeholder
                ="{{ __( 'Informe a descrição' ) }}">{{ $value ? $value->
                    description : null }}</textarea>
        </div>
    </div>
</div>

    <div class="field is-grouped">
        <div class="control">
            <button class="button is-link">{{ (int) $id > 0 ? __( 'Alterar' ) : __( '
                Cadastrar' ) }}</button>
        </div>
        <div class="control">
            <a href="{{ route('classifications.facets', ['classificationId' =>
                $classificationId]) }}" class="button is-text">{{ __( 'Cancelar' )
                }}</a>
        </div>
    </div>
</form>
</div>
</div>
@endsection

// resources/views/facet/values/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home' ) }}" aria-current="page">{{ __( 'Dashboard' ) }}</a></li>
            <li><a href="{{ route('classifications' ) }}" aria-current="page">{{ __( '
                Classificações' ) }}</a></li>
            <li><a href="{{ route('facets.edit', ['classificationId' => $classificationId, 'id
                ' => $facetId]) }}" aria-current="page">{{ $facet->title }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">Valores possíveis</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <div class="columns has-background-white">
        <div class="column is-12">
            <div class="tabs is-medium">
                <ul>
                    <li>
                        <a href="{{ route('facets.edit', [
                            'classificationId' => $classificationId,
                            'id' => $facetId,
                        ]) }}"><span>Definição</span></a>

```

```

        </li>
        <li class="is-active">
            <a href="#"><span>Valores possíveis</span></a>
        </li>
        <li>
            <a href="{ { route('facets.references', [
                'classificationId' => $classificationId,
                'id' => $facetId,
            ] ) }}"><span>Referências</span></a>
        </li>
    </ul>
</div>

<h2 class="subtitle">
    <a href="{ { route('facets.edit.values', [
        'classificationId' => $classificationId,
        'facetId' => $facetId,
        'id' => 0
    ] ) }}" class="button is-medium is-pulled-right">
        <span class="icon"><span class="mdi mdi-check"></span></span> <span>{{ __(
            'Novo Valor') }}</span>
    </a>
    &nbsp;
</h2>
<table class="table is-fullwidth is-striped is-hoverable is-bordered">
    <thead>
        <tr>
            <th style="width: 5%;"><abbr title="{ { __( 'Código') }}">#</abbr></th>
            <th style="width: 20%;">{{ __( 'Título') }}</th>
            <th>{{ __( 'Descrição') }}</th>
            <th>{{ __( 'Valor') }}</th>
            <th class="has-text-centered" style="width: 9%;">{{ __( 'Ações') }}</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($facetValues as $value)
            <tr>
                <td>{{ $value->id }}</td>
                <td>{{ $value->title }}</td>
                <td>{{ $value->description }}</td>
                <td>{{ $value->value }}</td>
                <td class="has-text-centered">
                    <a href="{ { route('facets.edit.values', [
                        'classificationId' => $classificationId,
                        'facetId' => $facetId,
                        'id' => $value->id
                    ] ) }}" class="button is-small">
                        <span class="icon"><span class="mdi mdi-pencil"></span></span></a>
                    <a class="button is-small delete-button"
                        data-target="modal" aria-haspopup="true" data-id="{ { $value->id
                        }}" data-classification=" "
                        onclick="event.preventDefault();">
                        <span class="icon"><span class="mdi mdi-delete"></span></span>
                    </a>
                </td>
            </tr>
        </tbody>
    </table>
</div>

<div class="modal" id="modal">
    <div class="modal-background"></div>
    <div class="modal-card">

```



```

<header class="modal-card-head">
  <p class="modal-card-title">{{ __( 'Remover valor' ) }}</p>
  <button class="delete modal-cancel-x" aria-label="close"></button>
</header>
<section class="modal-card-body">
  {{ __( 'Você tem certeza?' ) }}
</section>
<footer class="modal-card-foot">
  <button class="button is-danger modal-confirm" data-id="" data-classification=
    "" @click="attemptDeleteValue()">{{ __( 'Remover' ) }}</button>
  <button class="button modal-cancel">Cancelar</button>
</footer>
</div>
</div>
@endsection

// resources/views/facet_group/edit.blade.php

@extends('layouts.general')

@section('breadcrumb')
  <nav class="breadcrumb" aria-label="breadcrumbs">
    <ul>
      <li><a href="{{ route('home') }}" aria-current="page">{{ __( 'Dashboard' ) }}</a></li>
      <li><a href="{{ route('facets_groups') }}" aria-current="page">{{ __( 'Grupos de
        facetas' ) }}</a></li>
      <li class="is-active"><a href="#" aria-current="page">{{ (int) $id > 0 ? __( '
        Editar grupo' ) : __( 'Novo grupo' ) }}</a></li>
    </ul>
  </nav>
@endsection

@section('content')
  <div class="columns has-background-white">
    <div class="column is-12">
      <form id="form" action="{{ (int) $id > 0 ?
        route('facets_groups.update', ['id' => (int) $id]) :
        route('facets_groups.store') }}" method="POST">
        @if ((int) $id > 0)
          @method('PUT')
        @endif @csrf
        @if ($facetGroup) <input type="hidden" name="id" value="{{ $id }}"> @endif

      <div class="tabs is-medium">
        <ul>
          <li class="is-active">
            <a href="#"><span>Definição</span></a>
          </li>
        </ul>
      </div>

      <div class="box help-content has-background-white">
        <div class="field">
          <label class="label">{{ __( 'Título' ) }}</label>
          <div class="control">
            <input name="title" class="input is-medium" type="text"
              placeholder="{{ __( 'Informe o título' ) }}" value="{{
                $facetGroup ? $facetGroup->title : null }}">
          </div>
        </div>

        <div class="field">
          <label class="label">{{ __( 'Layout' ) }}</label>
          <div class="control">
            <div class="select is-info">
              <select name="layout">
                <option>Selecione um tipo de layout</option>
                <option @if ($facetGroup && $facetGroup->type === '
                  horizontal') selected @endif value="horizontal">
                  Horizontal</option>
                <option @if ($facetGroup && $facetGroup->type === '
                  vertical') selected @endif value="vertical">Vertical</

```



```

        <tfoot>
        <tr>
            <td colspan="4">
                {{ $facetsGroups->links('vendor.pagination.simple-default') }}
            </td>
        </tr>
        </tfoot>
    </table>
@endsection

// resources/views/layouts/partials/header.blade.php

<nav class="navbar is-white">
    <div class="container">
        <div class="navbar-brand">
            <a class="navbar-item brand-text" href="{{ route('home') }}">
                
            </a>
            <div class="navbar-burger burger" data-target="navMenu">
                <span></span>
                <span></span>
                <span></span>
            </div>
        </div>
        <div id="navMenu" class="navbar-menu">
            <div class="navbar-start">
                <a class="navbar-item" href="{{ route('home') }}">
                    {{ __('Dashboard') }}
                </a>
                <a class="navbar-item" href="{{ route('classifications') }}">
                    {{ __('Classificações') }}
                </a>
                <a class="navbar-item" href="{{ route('users') }}">
                    {{ __('Usuários') }}
                </a>
                <a class="navbar-item" href="/app" target="_blank">
                    {{ __('Ir para o guia') }}
                </a>
            </div>
            <div class="navbar-end">
                <a class="navbar-item" href="{{ route('logout') }}"
                    onclick="event.preventDefault();document.getElementById('logout-form').submit();">
                    {{ __('Sair') }}
                </a>
                <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
                    @csrf
                </form>
            </div>
        </div>
    </div>
</nav>

// resources/views/layouts/app.blade.php

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Painel administrativo') }}</title>

    <!-- Styles -->
    <link href="{{ asset('css/admin.css') }}" rel="stylesheet">
    <link href="{{ asset('css/admin.vendor.css') }}" rel="stylesheet">
    @yield('styles')

```

```

        @toastr_css
    </head>
    <body>
        <div id="app">
            @yield('body-content')
        </div>
        <!-- Scripts -->
        <script src="{{ asset('js/admin.js') }}" defer></script>
        <script src="{{ asset('js/admin.vendor.js') }}" defer></script>
        @yield('scripts')
        @jquery
        @toastr_js
        @toastr_render
    </body>
</html>

// resources/views/layouts/general.blade.php

@extends('layouts.app')

@section('body-content')
    <!-- Header navbar -->
    @include('layouts.partials.header')
    <!-- EOF Header navbar -->

    <div class="container is-fluid">
        <div class="columns">
            <div class="column is-12">
                <div class="container">
                    @yield('breadcrumb')

                    @yield('message')

                    @yield('content')
                </div>
            </div>
        </div>
    </div>
@endsection

// resources/views/layouts/login.blade.php

@extends('layouts.app')

@section('styles')
    <style type="text/css">
        html,body {
            font-family: 'Open Sans', serif;
            font-size: 14px;
            font-weight: 300;
        }
        .hero.is-success {
            background: #F2F6FA;
        }
        .hero .nav, .hero.is-success .nav {
            -webkit-box-shadow: none;
            box-shadow: none;
        }
        .box {
            margin-top: 5rem;
        }
        .avatar {
            margin-top: -70px;
            padding-bottom: 20px;
        }
        .avatar img {
            padding: 5px;
            background: #fff;
            border-radius: 50%;
            -webkit-box-shadow: 0 2px 3px rgba(10,10,10,.1), 0 0 0 1px rgba(10,10,10,.1);
            box-shadow: 0 2px 3px rgba(10,10,10,.1), 0 0 0 1px rgba(10,10,10,.1);
        }
        input {

```

```

        font-weight: 300;
    }
    p {
        font-weight: 700;
    }
    p.subtitle {
        padding-top: 1rem;
    }
}
</style>
@endsection

@section('body-content')
    <section class="hero is-success is-fullheight">
        <div class="hero-body">
            <div class="container has-text-centered">
                @yield('content')
            </div>
        </div>
    </section>
@endsection

// resources/views/user/edit.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li><a href="{{ route('users') }}" aria-current="page">{{ __('Usuários') }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ (int) $id > 0 ? __('Editar usuário') : __('Novo usuário') }}</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <div class="columns has-background-white">
        <div class="column is-8">
            <form id="form" action="{{ (int) $id > 0 ? route('users.update', ['id' => (int) $id]) : route('users.store') }}" method="POST">
                @if ((int) $id > 0)
                    @method('PUT')
                @endif
                @csrf
                <div class="field">
                    <label class="label">{{ __('Nome') }}</label>
                    <div class="control">
                        <input name="name" class="input is-medium" type="text" placeholder="{{ __('Informe o nome') }}" value="{{ $user ? $user->name : null }}" />
                    </div>
                </div>

                <div class="field">
                    <label class="label">{{ __('E-mail') }}</label>
                    <div class="control">
                        <input name="email" class="input is-medium" type="email" placeholder="{{ __('Informe o e-mail') }}" value="{{ $user ? $user->email : null }}" />
                    </div>
                </div>

                @if ((int) $id === 0)
                <div class="field">
                    <label class="label">{{ __('Senha') }}</label>
                    <div class="control">
                        <input name="password" class="input is-medium" type="password" placeholder="{{ __('Informe a senha') }}" />
                    </div>
                </div>
                @endif
            </form>
        </div>
    </div>

```

```

@endif

<div class="field is-grouped">
    <div class="control">
        <button class="button is-link">{{ (int) $id > 0 ? __('Alterar') : __('Cadastrar') }}</button>
    </div>
    <div class="control">
        <a href="{{ route('users') }}" class="button is-text">{{ __('Cancelar') }}</a>
    </div>
</div>
</form>
</div>
</div>
</div>
@endsection

// resources/views/user/index.blade.php

@extends('layouts.general')

@section('breadcrumb')
    <nav class="breadcrumb" aria-label="breadcrumbs">
        <ul>
            <li><a href="{{ route('home') }}" aria-current="page">{{ __('Dashboard') }}</a></li>
            <li class="is-active"><a href="#" aria-current="page">{{ __('Usuários') }}</a></li>
        </ul>
    </nav>
@endsection

@section('content')
    <h1 class="subtitle">
        <a href="{{ route('users.edit', ['id' => 0]) }}" class="button is-medium is-pulled-right">
            <span class="icon"><span class="mdi mdi-check"></span></span> <span>{{ __('Novo usuário') }}</span>
        </a>
        {{ __('Usuários') }}
    </h1>
    <table class="table is-fullwidth is-striped is-hoverable is-bordered">
        <thead>
            <tr>
                <th><abbr title="{{ __('Id') }}">#</abbr></th>
                <th>{{ __('Nome') }}</th>
                <th>{{ __('E-mail') }}</th>
                <th class="has-text-centered">{{ __('Ações') }}</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($users as $user)
                <tr>
                    <td>{{ $user->id }}</td>
                    <td>{{ $user->name }}</td>
                    <td>{{ $user->email }}</td>
                    <td class="has-text-centered">
                        <a href="{{ route('users.edit', ['id' => $user->id]) }}" class="button is-small">
                            <span class="icon"><span class="mdi mdi-pencil"></span></span>
                        </a>
                        <a class="button is-small delete-button" data-target="modal" aria-haspopup="true" data-id="{{ $user->id }}" data-classification="" onclick="event.preventDefault();">
                            <span class="icon"><span class="mdi mdi-delete"></span></span>
                        </a>
                    </td>
                </tr>
            @endforeach
        </tbody>
        <tfoot>
            <tr>

```

```

        <td colspan="4">
            {{ $users->links('vendor.pagination.simple-default') }}
        </td>
    </tr>
</tfoot>
</table>
<div class="modal" id="modal">
    <div class="modal-background"></div>
    <div class="modal-card">
        <header class="modal-card-head">
            <p class="modal-card-title">{{ __('Remover usuário') }}</p>
            <button class="delete modal-cancel-x" aria-label="close"></button>
        </header>
        <section class="modal-card-body">
            {{ __('Você tem certeza?') }}
        </section>
        <footer class="modal-card-foot">
            <button class="button is-danger modal-confirm" data-id="" data-classification=
                "" @click="attemptDeleteUser()">{{ __('Remover') }}</button>
            <button class="button modal-cancel">Cancelar</button>
        </footer>
    </div>
</div>
@endsection

// resources/views/vue/index.blade.php

<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
            minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <meta property="og:title" content="Guia facetado de técnicas elicitação de requisitos"/>
    <meta property="og:type" content="article"/>
    <meta property="og:description" content="Obtenha informações sobre técnicas de elicitação
        de requisitos!" />
    <meta property="og:image" content="http://retraining.inf.ufsc.br/guia/images/logo.png" />
    <meta property="og:url" content="http://retraining.inf.ufsc.br/guia"/>
    <meta property="og:site_name" content="REtraining - Guia facetado de Técnicas de
        Elicitação de Requisitos" />

    <meta name="twitter:card" content="summary">
    <meta name="twitter:url" content="http://retraining.inf.ufsc.br/guia">
    <meta name="twitter:title" content="Guia facetado de técnicas elicitação de requisitos">
    <meta name="twitter:description" content="Obtenha informações sobre técnicas de elicitação
        de requisitos!">
    <meta name="twitter:image" content="http://retraining.inf.ufsc.br/guia/images/logo.png">

    <base href="{{ config('app.base') }}">
    <title>REtraining - Guia facetado de Técnicas de Elicitação de Requisitos</title>

    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
    <link rel="stylesheet" href="{{ asset('css/vendor.css') }}">
</head>
<body>
    <div id="app">
        <loading :active.sync="this.$store.getters.isLoading" :can-cancel="false"></loading>

        <div class="columns">
            <sidebar></sidebar>
            <div class="column is-8 is-8-tablet is-9-desktop is-9-widescreen is-10-fullhd">
                <router-view ></router-view>
            </div>
        </div>
    </div>
    <script src="{{ asset('js/app.js') }}"></script>
    <script src="{{ asset('js/vendor.js') }}"></script>
</body>

```

</html>